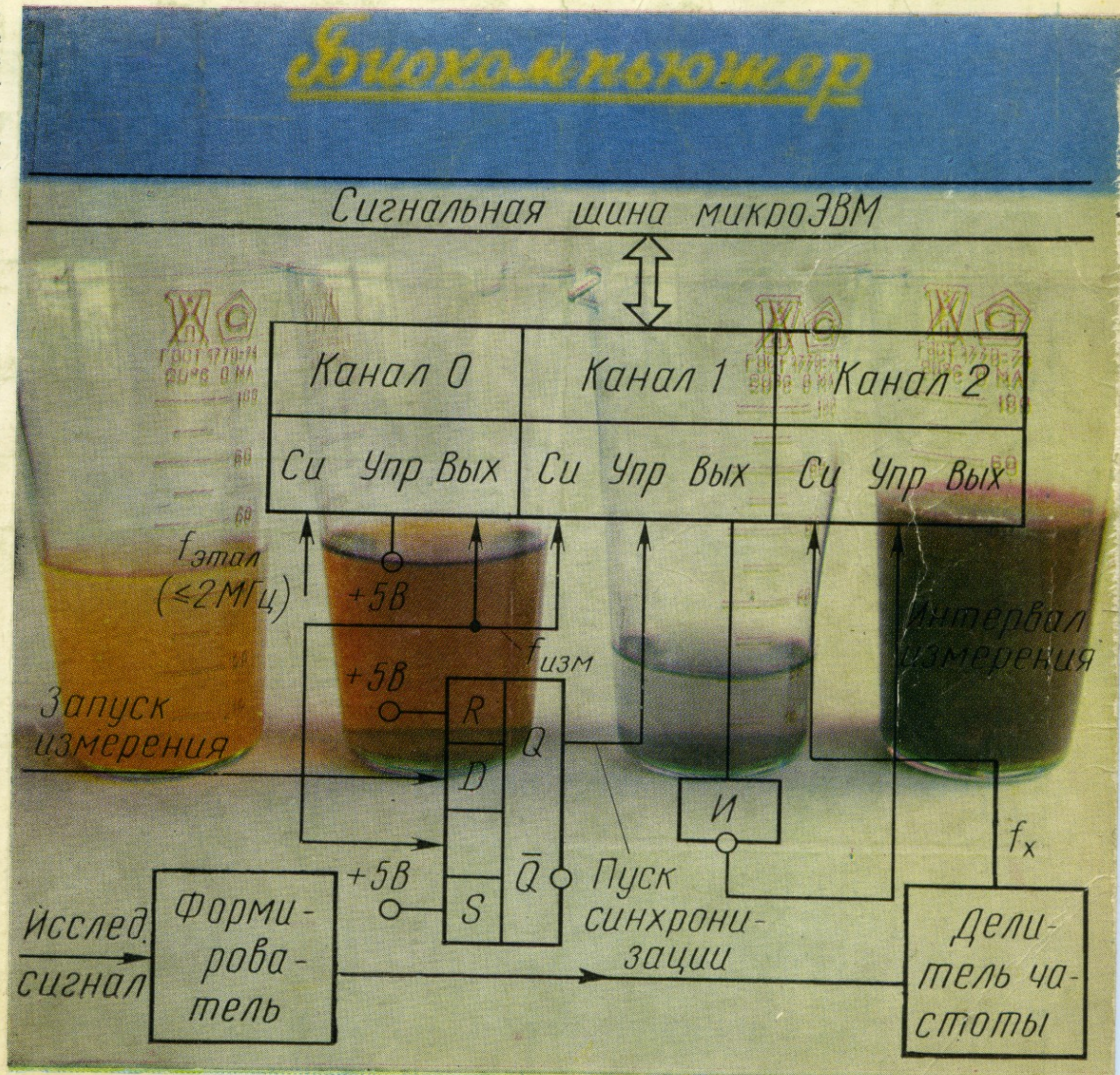
МИКРОПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

3 | 1984

ISSN 0233-4844

- **Программирование и искусственный интеллект:** создание и широкое использование баз знаний—актуальная задача ближайших десятилетий
- **Интеллектуальный интерфейс «человек—ЭВМ»,** разрабатываемый в ВЦ АН СССР, обеспечивает диалог для незнакомых с основами программирования пользователей в форме меню, таблиц, графиков и т. д.
- **От простейших видеомониторов до интеллектуальных дисплеев на основе матричных индикаторных панелей** — диапазон использования микропроцессоров и микроЭВМ в устройствах отображения информации
- **Интерфейсные микропроцессорные БИС серии К588** позволяют организовать мультиплексные каналы обмена данными в коде Манчестер-II
- **Система отладки микропроцессорных устройств «Электроника-Т3»** содержит комплект унифицированных плат схемных эмуляторов для различных типов 4-, 8-, и 16-разрядных микропроцессоров
- **Система управления транспортного робота** на основе микроЭВМ и БИС однородного процессора технологична и обладает повышенной надежностью
- **Биокомпьютеры** — перспективная элементная база индустрии обработки информации 1990—2000-х годов

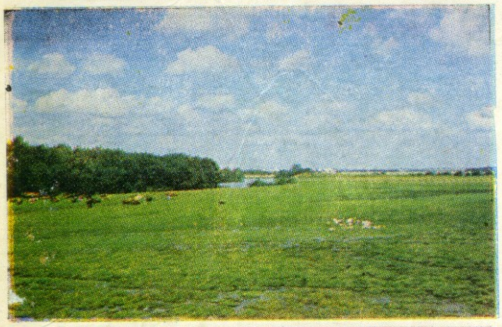


КОНТРОЛЛЕР ЗЕРКАЛЬНОЙ ФОТОКАМЕРЫ НА ОСНОВЕ БИС СЕРИИ КА 1808

(к ст. В. А. Бобкова, М. М. Боровницкого, В. В. Горового, В. А. Евдокимова)





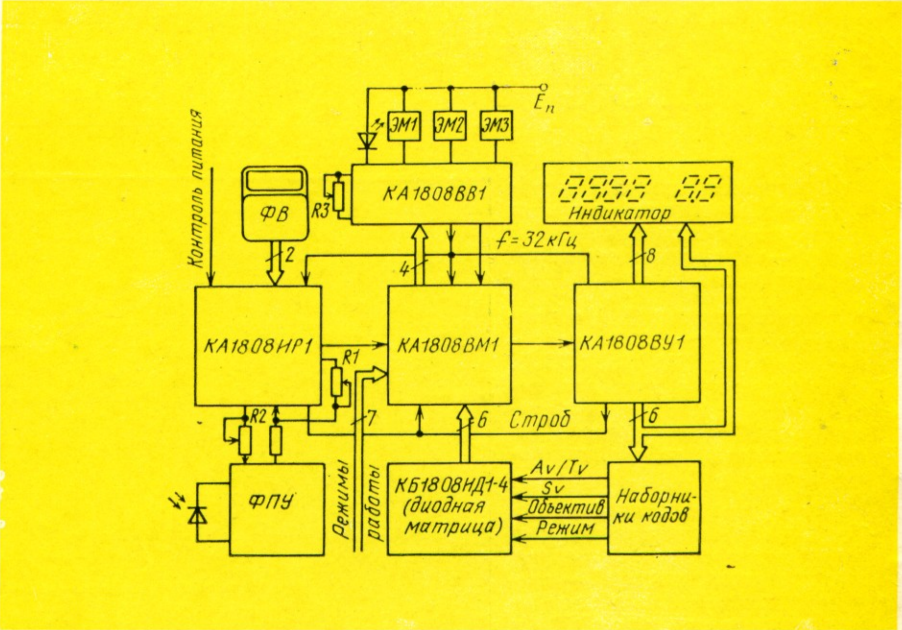






Контроллер зеркальной фотокамеры, созданный на основе четырех БИС микропроцессорного комплекта КА1808, полностью автоматизирует выбор диафрагмы и экспозиции (выдержки) — остается навести на резкость и нажать кнопку. Одинаковый уровень плотности отснятой фотопленки позволяет организовать ее групповую обработку.

Это значительно сокращает время получения готового фотоснимка, экономит фотореактивы и серебро.





Пролетарии всех стран, соединяйтесь!

ОРГАН ГОСУДАРСТВЕННОГО КОМИТЕТА СССР ПО НАУКЕ И ТЕХНИКЕ

Издается с 1984 года

МП МИКРОПРОЦЕССОРНЫЕ СРЕДСТВА И СИСТЕМЫ

ВЫХОДИТ ЧЕТЫРЕ РАЗА В ГОД НАУЧНО-ТЕХНИЧЕСКИЙ И ПРОИЗВОДСТВЕННЫЙ ЖУРНАЛ 3 | 1984 МОСКВА

СОДЕРЖАНИЕ

Адрес редакции: 101820, Москва, проезд Серова, 5, редакция журнала «Микропроцессорные средства и системы». Телефоны 228-18-88; 221-99-26

Сдано в набор 13.09.84
Подписано в печать 30.10.84
Формат 84×108 1/16. Бумага № 1.
Печать высокая Усл. печ. л. 10,08.
Уч.-изд. л. 15,1. Тираж 7 000 экз.
Заказ № 1340 Цена 1 руб. 10 коп.
Орган Государственного комитета СССР по науке и технике

Типография Всесоюзного центра информации по оборудованию ГКНТ СССР.
Московская типография № 13 Союзполиграфпрома Госкомиздата СССР

На первой странице обложки — Биокомпьютеры — фото О. В. Чиркина.

На второй странице обложки — Контроллер зеркальной фотокамеры на основе микропроцессорного комплекта БИС — фото В. В. Горового.

На четвертой странице обложки — Эффективность применения микропроцессоров в народном хозяйстве.

КОЛОНКА РЕДАКТОРА

Объем мирового производства микроЭВМ и микропроцессоров в начале 80-х годов превысил 100 млн. изделий в год. Функциональные характеристики и структурная сложность этих изделий существенно превосходят ЭВМ 50-х годов.

За точку отсчета эры ЭВМ обычно принимают начало сеансов опытной эксплуатации машины ENIAC в 1946 году. Напомним технические характеристики этой машины: масса — 30 т, число активных логических элементов (электронных ламп) — 18 тыс.; производительность — 5 тыс. операций/с. В начале 50-х годов насчитывалось около 100 машин сопоставимого класса, произведенных в США, СССР, Англии, Франции и ФРГ. К 1985 году суммарная структурная сложность ежегодно производимых средств вычислительной техники оценивается в 10^{14} активных логических элементов, что эквивалентно одной машине типа ENIAC на каждого жителя Земли.

Согласно прогнозам, к 2001 г. годовой прирост емкости запоминающих устройств ЭВМ будет равен общему числу нейронов мозга у всех 5,8 млрд. людей, которые, как ожидают, будут составлять тогда население нашей планеты.

Таким образом, экспоненциальный рост объема продукции информационной индустрии выводит мировой парк ЭВМ на уровень сложности, который оказывается сопоставимым с «венцом творения» — человеческим мозгом. Этот факт не может, видимо, не отражаться на научных методах, стиле, а в недалеком будущем и технологии разработки следующих поколений ЭВМ. Отдельным аспектам этой проблемы посвящены статьи В. Л. Дунина-Барковского и Н. Н. Всеволодова, которыми «МП» открывает впервые публикуемую в нашей стране серию статей по «биокомпьютерам».

Одновременно с ростом сложности ЭВМ неотвратимо приближается этап **гуманизации** *инженерной по своей природе профессии программиста. Некоторые из возникающих при этом научно-технических, социально-экономических и психологических проблем обсуждаются в статьях С. С. Лаврова, В. М. Брябрина, Г. С. Цейтина и других авторов этого номера «МП». Полемизируя на страницах «МП» с Г. С. Цейтиным, профессор кафедры психологии МГУ О. К. Тихомиров утверждает, что в настоящее время «выделение и решение психологических задач в разработке программного обеспечения представляется... не менее важной и интересной работой, чем сама разработка программного обеспечения». По-видимому, можно предполагать, что с ростом функциональной сложности ЭВМ ученым будет все труднее бороться с искушением использовать эту машину в качестве своего рода «зеркала» для попыток заглянуть внутрь интеллектуальной сущности человека.*

Совместными усилиями создателей средств микропроцессорной техники и разработчиков интеллектуальных систем программирования ЭВМ превращается из загадочного монстра, с которым люди могли общаться только через представителей узкой профессиональной касты жрецов, называющих себя прикладными программистами, в простой и понятный индивидуальный инструмент, доступный для повседневной эксплуатации в быту, на работе, в учебе... — во всех без исключения сферах человеческой деятельности.

Г. Р. Громов

УДК 681.3.06

С. С. Лавров

ИСПОЛЬЗОВАНИЕ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ, ПРОГРАММИРОВАНИЕ И ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ (перспективы развития)

Введение

В данной статье рассматривается место и роль вычислительной техники в современном обществе и пути наиболее эффективного использования этой техники в интересах общества.

Между обществом и вычислительной техникой можно выделить, по меньшей мере, два промежуточных звена: задачи и программы. Задачи возникают как средство достижения целей, стоящих перед обществом, программы — как средство заставить вычислительную машину решить поставленные задачи. Отсюда ясен приоритет общественных *целей* перед конкретными *задачами*, задач — перед *программами* и программ — перед *апаратными средствами* вычислительной техники. Последнее проявляется, в частности, в том, что на разработку программ расходуется в два с лишним раза больше средств, чем на разработку аппаратуры.

Естественно, что имеются в виду лишь те задачи, которые прямо связаны с обработкой информации, так как только с информацией имеют дело вычислительные машины. В связи со значительным усложнением социально-экономической структуры общества в период научно-технической революции, появлением совершенно новых отраслей промышленности и видов продукции, рождением новых научных дисциплин и возрастанием роли науки в жизни общества, в связи с усложнением технологических процессов, самих изделий и роста числа параметров, по которым они оцениваются, роль информации, информационных процессов в жизни общества неизмеримо возросла. Этим, по существу, и вызвано появление вычислительной техники и необычайно высокий темп роста ее применений. Вторая причина — большие явные и скрытые возможности вычислительной техники, бурный рост этих возможностей, пределов которому пока не видно, универсальность и гибкость средств вычислительной техники.

Если задачи возникают непосредственно в обществе, причем на всех уровнях общественной иерархии, то программы часто воспринимаются как помеха на пути использования вычислительной техники, как неизбежное зло. Это вызвано тем, что программы сами по себе нужны не человеку, а лишь машине, чтобы она, работая по программе, выработала результат решения задачи. Он-то и является конечным желаемым звеном всей цепочки. Впрочем, результат решения одной задачи обычно оказывается лишь отправной точкой для постановки новых задач.

Центральная проблема в области использования вычислительной техники — это сокращение усилий, времени и средств, затрачиваемых на программирование, обеспечение возможности сосредоточиться на самой задаче. Для этого нужно, чтобы машина непосредственно «понимала» задачу, поставленную человеком, а для этого надо научиться передавать машине человеческие знания.

Есть еще одна, не менее важная причина, по которой следует переходить от традиционных средств программирования к средствам более высокого уровня, ориентированным непосредственно на задачи, а не на программы. Проектируемые супер-ЭВМ будут способны выполнять программы, в десятки и сотни раз более сложные, чем самые большие программы для современных ЭВМ. Но уже сейчас составление больших программ традиционными средствами находится на пределе человеческих возможностей. Надо искать и внедрять новые средства, позволяющие автоматизировать не только трансляцию и выполнение, но и разработку, синтез алгоритмов непосредственно по постановке задачи.

Такое повышение уровня средств общения человека с вычислительной техникой возможно только на основе интенсивного использования искусственного интеллекта. Необходимо также развивать существующие средства программирования и системное программное обеспечение ЭВМ.

Центральная проблема в области использования вычислительной техники — это сокращение усилий, времени и средств, затрачиваемых на программирование, обеспечение возможности сосредоточиться на самой задаче. Для этого нужно, чтобы машина непосредственно «понимала» задачу, поставленную человеком, а для этого надо научиться передавать машине человеческие знания.

1. Виды знания и базы знаний

Как отмечалось выше, чтобы машина понимала задачу, она должна обладать запасом знаний. Знания человека, в том числе — знания, передаваемые им машине, делятся на три основные категории.

Концептуальное, понятийное знание воплощено, прежде всего, в словах человеческой речи, конкретнее — в научно-технических терминах и, естественно, в стоящих за этими терминами классах объектов окружающего нас мира и их свойствах. Сюда входят также связи, отношения и зависимости между понятиями и их свойствами, причем связи абстрактные, также выражаемые словами и терминами. Концептуальное знание — это сфера, главным образом, фундаментальных наук, но и прикладные науки во многом базируются на этой форме знания.

Фактуальное, предметное знание — это совокупность сведений о качественных и количественных характеристиках конкретных объектов. Именно с этой категорией знания чаще всего связываются термины *«информация»* и *«данные»*, хотя такое использование этих терминов несколько принижает их значение. Любое знание несет в себе информацию и может быть представлено в виде данных. Фактуальное знание — это то, с чем прежде всего имели дело вычислительные машины и с чем они больше всего имеют дело по настоящее время. Современная форма накопления фактуального знания — это то, что принято называть базами данных. Правда для организации баз данных, для поиска в них нужной информации надо опираться на концептуальное знание.

Алгоритмическое, процедурное знание — это то, что принято называть словами «умение», «технология», «know-how» и т. п. В вычислительном деле алгоритмическое знание реализуется в виде алгоритмов, программ и подпрограмм, но не всяких, а таких, которые могут передаваться из рук в руки и использоваться без участия их авторов. Такая реализация алгоритмического знания называется программным продуктом. Наиболее распространенная, но не единственная форма программного продукта — это пакеты прикладных программ. Они обычно ориентированы на конкретную область применения вычислительной техники. Организация и использование пакетов прикладных программ также базируется на концептуальном знании.

Ясно, что концептуальное знание является более высокой, определяющей категорией знания, хотя, с точки зрения практики, другие категории могут казаться более важными.

...концептуальное знание редко воплощается в форме, доступной для обработки на ЭВМ. А если воплощается, то чаще всего неполно и односторонне. Носителем концептуального знания в большинстве случаев остается человек. Это тормозит автоматизацию многих процессов.

Вероятно, именно по последней причине концептуальное знание редко воплощается в форме, доступной для обработки на ЭВМ. А если воплощается, то чаще всего неполно и односторонне. Носителем концептуального знания в большинстве случаев остается человек. Это тормозит автоматизацию многих процессов.

Системы машинного представления концептуального знания, а точнее — системы, реализующие все три категории знания, но выделяющие концептуальное знание на первый план и работающие на основе его интенсивного использования, называются базами знаний. Их можно называть и системами искусственного интеллекта, хотя для последних присутствие двух других категорий знания не столь существенно и даже не обязательно — они и получают задание и дают ответ в терминах только концептуального знания.

Создание и широкое использование баз знаний во всех областях применения вычислительной техники — одна из актуальнейших задач ближайших десятилетий.

Ниже будем придерживаться следующей терминологии. Концептуальную часть базы знаний будем называть моделью предметной области, алгоритмическую часть — пакетом прикладных программ, а фактуальную часть — базой данных. Эта терминология практически совпадает с общепринятой.

2. Задачи и спецификации

Задача может быть решена машиной только в том случае, если она формально поставлена — если для нее написана формальная спецификация. Последняя должна опираться на некоторую базу знаний. Модель предметной области описывает общую обстановку, в которой возникла задача, а спецификация — содержание задачи. Их сочетание позволяет установить, какие абстрактные связи и зависимости, в каких сочетаниях и в какой последовательности должны быть использованы для решения задачи. Пакет прикладных программ представляет конкретные программные средства, стоящие за этими зависимостями, а также содержит алгоритмы, которые могут быть использованы для решения возникающих при этом уравнений и систем уравнений. Наконец, база данных поставляет все или часть исходных данных для выполнения этих алгоритмов, недостающие данные должны содержаться в спецификации.

Этим трем частям базы знаний соответствуют три этапа решения задачи: нахождение абстрактной программы решения задачи (в терминах концептуального знания), перевод этой программы на доступный машине язык (не обязательно машинный язык, а возможно, один из языков программирования в традиционном смысле слова), дальнейшая трансляция и выполнение программы. Пионером такого подхода к использованию ЭВМ является член-корреспондент АН Эстонской ССР Э. Х. Тыугу.

Построение абстрактной программы — это задача, лежащая в рамках искусственного интеллекта и решаемая его методами. Перевод абстрактной программы на алгоритмический язык — задача более традиционная для (системного) программирования. Она решается средствами и методами теории трансляции, в частности, — средствами макрообработки. Последний этап реализуется вполне привычными средствами, существующими уже сейчас (в основном).

Описанная схема содержит некоторые пробелы.

Поясним один из них на примере самолетостроения в качестве предметной области. Модель этой предметной области содержит общие понятия, относящиеся к конструированию самолетов, а также понятия смежных областей знания — аэродинамики, теории упругости, материаловедения и т. д. Но этого еще недостаточно для постановки задач, связанных с проектированием конкретного самолета. Должны быть как-то определены тип самолета, основные требования к нему — значения его главных проектных характеристик, ограничения на параметры, выбираемые в ходе проектирования, возможные варианты конструкции и т. п. Все это вместе составляет описание объекта, по отношению к которому ставится задача.

В ходе решения основной задачи возникают различные частные задачи, например, аэродинамический расчет самолета. Они могут распадаться на подзадачи — расчет максимальной дальности полета в зависимости от полезной нагрузки, определение маневренности, длины пробега при посадке и т. п.

Подобные задачи и подзадачи, относящиеся к одному объекту, могут ставиться и решаться независимо друг от друга разными людьми. Их решение может приводить к изменению, уточнению описания объекта, тогда как модель предметной области остается неизменной (если только в ходе решения задачи не было сделано открытий фундаментального характера — например, обнаружен новый вид динамической неустойчивости самолета и найдены методы ее расчета и борьбы с ней).

Описания исследуемых объектов составляют текущую, динамическую часть базы знаний, занимающую промежуточное положение между моделью предметной области и базой данных, ближе к последней. Такое описание неправомерно считать частью спецификации задачи, поскольку по отношению к одному объекту ставится, как правило, много задач.

Второе упрощение, содержащееся в изложенной выше схеме, — это допущение, что задача может быть автоматически решена до конца, как только для нее составлена формальная спецификация. В действительности, спецификации — сложные объекты, требующие проверки их правильности, отладки и исправления в ходе решения задач. Поэтому решение задач может быть лишь автоматизированным — оно требует постоянного участия человека и тщательного контроля с его стороны. Автоматизированные системы решения задач должны быть поэтому диалоговыми. Они должны очень точно информировать человека о всех трудностях, возникших в ходе решения задачи. Наиболее вероятный источник этих трудностей — неполнота спецификации задачи или описания объекта. Не исключена также неполнота базы знаний, особенно, на

Системы машинного представления концептуального знания... называются базами знаний. Создание и широкое использование баз знаний во всех областях применения вычислительной техники — одна из актуальнейших задач ближайших десятилетий.

первых порах ее использования. Вполне возможно также, что перед системой поставлена объективно трудная задача, выходящая за рамки ее возможностей.

Для контроля за правильностью решения система должна снабжать человека достаточно полной информацией о промежуточных шагах решения, о полученных при этом промежуточных результатах.

Изложенная схема представляет собой типичный, универсальный метод использования баз знаний. Создание и внедрение автоматизированных диалоговых систем решения задач на основе баз знаний следует считать одной из главных проблем использования средств вычислительной техники.

Задача может быть решена машиной только в том случае, если она формально поставлена — если для нее написана формальная спецификация. Последняя должна опираться на некоторую базу знаний.

Модель предметной области описывает общую обстановку, в которой возникла задача, а спецификация — содержание задачи.

Надо ориентироваться на постепенный плановый переход от алгоритмических языков к языкам спецификаций в качестве основы использования средств вычислительной техники.

3. Языковые проблемы

Наиболее распространенные и известные языки программирования (Фортран, Алгол, Кобол, ПЛ-1, Алгол-68, Паскаль, Ада и др.) принадлежат фактически к одному семейству.

Применение языков этого семейства обеспечивало на протяжении более чем 25 лет достаточно эффективное использование вычислительной техники. За это время это семейство языков проделало заметную эволюцию, но в основе оно осталось тем же, чем было с самого начала — классом языков операторного, процедурного типа. Это — типичные языки для описания и представления алгоритмического знания.

Более разнообразны языки, ориентированные на описание и использование фактуального знания. Они лет на 10 моложе. Их уровень в некотором отношении выше уровня языков первого семейства — они ближе по своей природе к языкам спецификаций. Они описывают, с одной стороны, структуру базы данных, затрагивая ее концептуальную сторону (имена таблиц, доменов и атрибутов — это, как правило, содержательные термины). С другой стороны, они предоставляют средства для формулировки запросов к базе данных также на более или менее содержательном уровне, не касаясь способа получения ответа на запрос. Особенно характерны в этом отношении реляционные базы данных, получившие, впрочем, наименьшее распространение.

И, наконец, существует семейство языков, ведущих, в основном, свое начало от языка Лисп (1960 г.). Это — различные версии Лиспа, а также языки Плэннер, Коннайвер и др. Примыкает к ним язык Пролог, интерес к которому особенно возрос после того, как японцы в своем проекте ЭВМ пятого поколения отвели Прологу одно из центральных мест. Это семейство характеризуют поразному — как языки для обработки символьной информации, языки искусственного интеллекта, языки управления роботами, языки логического программирования или же как функциональные, непроцедурные языки. Некоторые их черты, такие как списки свойств и ассоциативные списки, прямо ориентированы на предоставление знаний, другие — использование списочных структур, рекурсивность, механизм возвратов (бектрекинга) — на решение логических задач типа автоматического доказательства теорем. Все это крайне важно для создания и использования баз знаний.

Обратимся к другим классам формальных языков, лежащим вне сферы программирования. Почти любая область человеческой деятельности вырабатывает для себя, если не полностью формальный или формализованный язык (математические и химические формулы, технические чертежи, шахматная нотация, нотная грамота и т. д.), то весьма специфический жаргон естественного языка. Естественный язык хорош для повседневного общения и явно неудобен, недостаточен (даже в общении между людьми) там, где требуется точность, определенность и лаконичность — особенно в математике, естественных науках, технике.

Можно с уверенностью утверждать, что естественный язык непригоден в подавляющем большинстве случаев общения человека с вычислительной техникой. Он хорошо приспособлен для восприятия человеком с его высоким уровнем интеллекта и богатейшими возможностями ассоциативного восприятия, но крайне сложен для анализа и правильного понимания машиной, «интеллект» которой весьма специфичен, а ассоциативные возможности, даже для будущих поколений, — очень ограничены. Пока число параллельно работающих процессоров вычислительного комплекса не достигнет 10^6—10^8, а число битов оперативной памяти 10^{12}—10^{15}, трудно надеяться, что такой комплекс сможет достаточно эффективно воспринимать человеческую речь. Исключение может быть сделано для самых простых вопросов и заданий, почти не допускающих двоякого толкования, но и здесь вероятность ошибочной реакции остается очень большой (порядка нескольких процентов).

С другой стороны, очень желательно, чтобы машина формулировала свои сообщения человеку на языке, близком к естественному, особенно — в диалоговых системах, в нестандартных ситуациях. Это не очень сложно осуществить, и во многих существующих системах так и делается.

Среди формальных языков особенно выделяются языки математической логики, а среди них — язык (классического) исчисления предикатов первого порядка. Вот уже около 100 лет он используется в математике как (абстрактный) язык представления знаний и за это время он не претерпел существенных изменений. На нем в большей или меньшей степени базируются все другие логико-математические языки, а также многие средства языков программирования, особенно — языков последнего из названных выше семейств.

Экспериментальные системы логического вывода (программы автоматического доказательства теорем) также, большей частью, используют этот язык, иногда с некоторыми ограничениями, реже — в расширенном виде.

Пригодность этого языка в качестве основы для практического языка представления и использования знаний вызывает, тем не менее, бурные споры. Возражения против него часто бывают основаны на недостаточном опыте его использования в практических целях. С другой стороны, нет сомнения, что языки представления знаний еще должны пройти достаточно долгий путь развития. Но при этом важно, чтобы отправная точка была выбрана правильно.

Язык представления знаний и спецификаций задач неоднороден — он должен состоять из языковых средств разного назначения и разной степени сложности, предназначенных для описания:

— моделей предметных областей (концептуального знания);

— связей модели предметной области с поддерживающими ее пакетом прикладных программ и базой данных;

— объектов, по отношению к которым ставятся задачи;

— собственно спецификаций задач.

Наиболее сложны средства первой

категории, наиболее просты средства спецификации задач, ориентированные на массового пользователя. Соответственно должна убывать степень формализованности этих средств, увеличиваться их внешняя близость к естественному языку. Но по существу все эти средства должны оставаться строго формальными и однозначно понимаемыми.

Из сказанного вытекают следующие задачи в области развития средств использования вычислительной техники: разработка и развитие языковых средств представления знаний и спецификаций задач, исследование возможностей и развитие методов общения человека с машиной на естественном языке, исследования в области формально-логических языков с целью выработки основы для наиболее естественного и полного представления и использования знаний.

4. Искусственный интеллект

Выше, в разделе «Виды знания и базы знаний», искусственный интеллект был охарактеризован как система представления и обработки концептуального знания. По-видимому, это наиболее общее и точное его определение. Часто искусственный интеллект связывают с обработкой текстов и устных сообщений на естественном языке. Поскольку естественный язык — это форма представления знаний, наиболее широко используемая человеком, то обработка сообщений на естественном языке должна составлять важную область исследований по искусственному интеллекту, но далеко не единственную.

Более конкретно, задача обработки сообщений на естественных языках распадается на следующие подзадачи:

— анализ и синтез сообщений в письменной и устной форме;

— преобразование сообщений во внутреннее представление и обратно;

— перевод с одного естественного языка на другой.

Отдельно следует упомянуть задачу автоматического реферирования текстов на естественных языках.

Анализ и обработка (распознавание) всех видов изображений также тесно примыкают к сфере искусственного интеллекта, но не исчерпываются им, так как изображения, особенно — картины внешнего мира, обычно содержат очень много сверх того, что может быть отнесено к категории концептуального знания. Лишь после того как из изображения вычленены элементы, несущие содержательную смысловую нагрузку (в рамках поставленной задачи), дальнейший анализ относится к области искусственного интеллекта.

Автоматическое доказательство теорем — это одна из центральных проблем искусственного интеллекта, но она также не исчерпывает его содержания. Почти все задачи естественным путем сводятся к доказательству некоторой теоремы, а именно — теоремы существования решения задачи. Но до этапа доказательства надо осуществить такое сведение, а после доказательства теоремы — извлечь из него способ (алгоритм) получения решения.

В явном или неявном виде доказательство теоремы существования всегда присутствует в решении задачи и обычно оказывается самым сложным этапом решения. Именно поэтому столь велика роль автоматического доказательства теорем в проблеме искусственного интеллекта.

В явном или неявном виде доказательство теоремы существования всегда присутствует в решении задачи и обычно оказывается самым сложным этапом решения. Именно поэтому столь велика роль автоматического доказательства теорем в проблеме искусственного интеллекта.

Почти все созданные к настоящему времени отечественные и зарубежные программы автоматического доказательства теорем работают в рамках исчисления предикатов первого порядка. Очень многие из них базируются на методе резолюций, хотя в Советском Союзе были заложены основы более мощного и универсального подхода — обратного метода С. Ю. Маслова.

Метод резолюций предполагает существенное преобразование исходной формы теоремы. Это затрудняет для человека слежение за поиском доказательства, мешает разобрать структуру доказательства. Необходимо развивать другие методы, в частности — упомянутый выше обратный метод, сохраняющие исходную форму теоремы и посылок, из которых она выводится.

Следует также развивать методы доказательства теорем в других логических исчислениях — логиках высших порядков, модальных логиках, динамических логиках и т. д. Классы логических исчислений следует отбирать, руководствуясь системой представления концептуального знания в базах знаний, с тем чтобы программы доказательства теорем не были излишне сложными.

Более того, для многих предметных областей характерны специфические приемы доказательства теорем существования. Включение описания таких приемов в модель предметной области с соответствующей алгоритмической поддержкой может очень существенно повысить эффективность систем искусственного интеллекта и базирующихся на них систем решения задач. Это ярко продемонстрировали работы, проводимые под руководством член-корреспондента АН СССР В. М. Матросова.

Итак, насущные проблемы в области искусственного интеллекта сводятся к следующему:

— развитие методов и алгоритмов автоматического доказательства теорем в различных логических исчислениях;

— поиск специфических методов и алгоритмов доказательства теорем для конкретных предметных областей;

— использование автоматического доказательства теорем в автоматизированных системах решения задач;

— развитие методов и алгоритмов обработки письменных и устных сообщений на естественных языках;

— развитие методов автоматического реферирования текстов на естественных языках;

— развитие методов анализа, распознавания, синтеза и преобразования изображений.

5. Автоматизированное проектирование ЭВМ

Выше шла речь об автоматизированных системах решения задач на ЭВМ. На выходе таких систем получается программа решения задачи, которая затем выполняется, как любая другая программа.

Параллельно с такими системами будут создаваться автоматизированные системы проектирования ЭВМ (или СБИС) под заданную программу ее работы. Это может быть некоторая управляющая программа, подпрограмма операционной системы, программа трансляции с некоторого языка и т. п.

На некотором этапе развития таких систем окажется целесообразным их объединение в единую систему автоматизированного проектирования специализированной ЭВМ для решения задачи, представленной своей спецификацией. Обе части такой системы — проектирование алгоритма по спецификации и проектирование ЭВМ по алгоритму — должны строиться с учетом их взаимодействия. Не исключено, что удастся избавиться от алгоритма, как промежуточного звена проектирования. При этом исходная спецификация будет постепенно порождать ряд промежуточных, более простых спецификаций, пока не будут получены

спецификации столь простые, что не составит труда построить функциональную схему аппаратуры, решающей соответствующую задачу. Затем эти схемы должны быть объединены в одну, соответствующую исходной задаче. После оптимизации объединенной схемы начинается более рутинный этап — конструирование аппаратуры под найденную функциональную схему.

Исследование теоретических основ, методов и технологии проектирования ЭВМ (СБИС, аппаратуры) по спецификации задачи через некоторое время должно стать одной из центральных проблем развития средств вычислительной техники.

6. Традиционные средства программирования

Используемые в настоящее время средства общения человека с вычислительной техникой и компоненты системного программного обеспечения будут и в дальнейшем широко использоваться как самостоятельно, так и во взаимодействии со средствами более высокого уровня. Они должны развиваться и совершенствоваться в следующих основных направлениях.

Системное программное обеспечение в целом должно стать более сбалансированным, построенным на единой концептуальной основе, а также более эффективным и надежным. Необходимо значительно глубже анализировать ситуации, возникающие в ходе вычислительного процесса, и повысить информативность всех диагностических сообщений, чтобы пользователь мог скорее осознать существо допущенных им ошибок и найти пути их исправления. Формальная спецификация задачи и здесь может принести большую пользу.

Операционные системы представляют собой средства управления вычислительным комплексом и его компонентами, реализованные программным путем. Должны быть систематизированы и сделаны более общими, фундаментальными принципы построения и функционирования операционных систем. Настоятельно необходима унификация, по крайней мере — на внешнем уровне, архитектур операционных систем для различных типов и семейств вычислительной техники. Должен быть изучен и обобщен положительный опыт разработки и внедрения таких операционных систем, как ОС МВК «Эльбрус», а также UNIX и других операционных систем малых ЭВМ. Должна получить дальнейшее развитие виртуализация всех ресурсов ЭВМ. На этой основе операционные системы должны стать более скрытыми от пользователя, требовать от него меньше информации, которую он в настоящее время сообщает, как правило, наугад.

Алгоритмические языки (процедурного типа) также должны развиваться в сторону фундаментализации их основных понятий и поиска новых эффективных изобразительных средств. В эти языки должно прочно войти понятие абстрактного типа данных. Должны быть усовершенствованы средства организации параллельных процессов и их взаимодействия и развиты средства автоматического распараллеливания программ. Должны получить развитие средства управления процессами, такие как итераторы, сопрограммы и др. Данные в языках должны быть строго типизированы. Однако наряду со статическим контролем типов должен получить распространение динамический контроль типов значений переменных, особенно при условии аппаратной поддержки (по образцу автокода Эль-76 для МВК «Эльбрус»).

Алгоритмические языки должны стать единым и универсальным средством общения человека с вычислительным комплексом, вобрать в себя средства связи с операционной системой (средой), с редакторами связей и загрузчиками, средства организации диалога, перехода в диалоговый режим и выхода из него, средства установления контакта с другими программами и пользователями, средства макрообработки и т. п.

Алгоритмические языки должны стать единым и универсальным средством общения человека с вычислительным комплексом, вобрать в себя средства связи с операционной системой (средой), с редакторами связей и загрузчиками, средства организации диалога, перехода в диалоговый режим и выхода из него, средства установления контакта с другими программами и пользователями, средства макрообработки и т. п.

Средства макрообработки должны стать существенно более эффективными и мощными. В то же время они должны поддерживать структурный стиль и дисциплину программирования.

Большое развитие должны получить диалоговые средства общения пользователя с вычислительным комплексом в связи с тем, что диалог становится основным режимом использования вычислительной техники. Переходы от этапа редактирования программы и данных к этапу выполнения программы и этапу просмотра результатов должны стать более простыми и естественными. Программа и данные должны быть тщательно защищены от ошибок пользователя, процесс их восстановления должен быть упрощен. Необходимо предусмотреть возможность прерывания и возобновления диалога без потери связей, установленных в предыдущем сеансе.

В диалоге должна широко использоваться полиэкранная техника (в пределах одного или нескольких физических экранов). Соответственно должны быть развиты средства редактирования и просмотра входных и выходных данных.

Локальные сети с персональными ЭВМ в качестве терминалов должны стать основным средством коллективной работы пользователей. Необходимо разработать эффективные средства оперативного общения (совместной работы) двух и более пользователей, групповые архивы с хорошей защитой информации от несогласованных действий пользователей.

В программном обеспечении профессиональных персональных ЭВМ должны получить развитие такие средства как высокоавтоматизированная система ведения лабораторного журнала, быстрого запуска типичных задач, индивидуальные базы данных с гибкой структурой, средства оргтехники и т. п.

7. Человеческие аспекты

Люди, использующие вычислительную технику, никогда не будут в одинаковой степени владеть этой техникой, знать ее возможности и методы ее использования. Да это и не требуется. Однако очень важно, чтобы в любой области, где применяются средства вычислительной техники, нашлись люди высокой квалификации не только в своей области, но и в области использования этих средств. Только таким специалистам под силу создание модели данной предметной области и ее программной поддержки — пакетов прикладных программ и баз данных. Безусловно, в этой работе, особенно во второй ее части, должны участвовать и программисты-профессионалы. Но последние самостоятельно не смогут создать модель предметной области, где требуются, в первую очередь, все же профессиональные знания, а во вторую — не столько программистская, сколько математическая культура — умение абстрактно мыслить и пользоваться формально-языковыми средствами. Иначе говоря, разработчики моделей предметных областей — это носители фундаментального, системного знания в своей области.

Соответственно, специалистов, создающих описание объекта исследования, можно назвать носителями конструктивного знания. Им также программистские знания еще не очень нужны, от них требуется изобретательский дар, фантазия, умение мысленно построить такой объект, какого еще не существовало в данной области и который, возможно, будет отвечать поставленной цели. В какой мере и что нужно изменить, чтобы цель была достигнута, — должно показать исследование этого объекта, решение различных задач, к нему относящихся, с использованием базы знаний и всех возможностей вычислительной техники.

...специалистов, создающих описание объекта исследования, можно назвать носителями конструктивного знания... от них требуется изобретательский дар, фантазия, умение мысленно построить такой объект, какого еще не существовало в данной области и который, возможно, будет отвечать поставленной цели.

Наконец, все остальные специалисты — самая многочисленная часть коллектива — ставят конкретные задачи по исследованию объекта, поручая поиск решения системе автоматического решения задач.

От них требуется пассивное владение моделью предметной области — умение прочитать ее описание и сформулировать задачу так, чтобы эта формулировка — спецификация задачи — согласовалась с описанием модели. Менять это описание они не вправе. Более активно они должны владеть описанием объекта. Менять его по существу они сами также не могут, но они должны определить или уточнить значения каких-то его параметров, а возможно, — сделать это описание более развернутым, более детальным (в своей части).

Такое распределение обязанностей по исследованию или проектированию объекта трébует как говорилось в разделе «Языковые проблемы», разного уровня языков, которыми пользуются специалисты. Оно требует также, чтобы разные специалисты и готовились по-разному, получали разную квалификацию. Это не просто более высокая или более низкая квалификация, речь идет о разном характере деятельности. Различие квалификаций обеспечивается, прежде всего, различием вузов и техникумов, факультетов и кафедр, где готовятся специалисты. Но этого мало. Помимо своей основной профессиональной подготовки, специалисты должны приобрести знания в смежных областях — в математике или программировании, в электронике или приборостроении и т. п.

Сейчас во всех учебных заведениях вплоть до ведущих университетов страны на фоне роста числа специальностей проводится линия на унификацию образования в пределах данной специальности, на сокращение факультативных форм обучения. Делается это под благовидным предлогом — ликвидировать специальные курсы и семинары с малым количеством учащихся, увеличить тем самым коэффициент использования, эффективность работы преподавателей, профессоров и доцентов. В результате смежные дисциплины преподаются всем учащимся, но в очень ограниченном, урезанном объеме. Возможности учащихся получить образование, соответствующее их индивидуальным склонностям и способностям, резко ограничиваются, а перегрузка учащихся растет.

Было бы значительно целесообразнее давать учащимся подготовку в области всего одной-двух смежных дисциплин, но более глубокую. Во всех видах образования должен быть обеспечен единый минимум общепрофессиональной подготовки. Дополнительное образование в смежных областях знания должно стать значительно более факультативным. Для этого должны организовываться (и очень тщательно планироваться) межфакультетские, межкафедральные курсы, семинары и занятия. При этом должны, естественно, учитываться потребности общества в специалистах разного профиля (включая дополнительную специализацию), загрузка преподавателей и учащихся и их индивидуальные интересы.

Любая система образования сталкивается с трудно разрешимым противоречием. С одной стороны образование должно быть универсальным, чтобы выпускник был подготовлен к любому виду деятельности — производственной, научной, организаторской — которой ему придется заниматься, а также, чтобы он, в случае необходимости, мог легко сменить место работы, вид занятий и даже профессию. С другой стороны, требуется удешевить и ускорить подготовку специалистов, чтобы они раньше смогли начать трудовую деятельность.

Первое обстоятельство вынуждает расширять и насыщать программы, продлевать сроки обучения, отодвигать начало специализации учащихся. При этом для многих из них разнообразие и насыщенность учебного материала, количество и существо требований к студентам оказываются или кажутся непосильными, что резко сказывается на отношении к учебе и на качестве знаний.

Второе обстоятельство заставляет множить число вузов, факультетов, кафедр и специальностей. Внешне система высшего образования представляется достаточно гибкой и разнообразной. Но по существу молодой специалист редко оказывается вполне подготовленным к началу трудовой деятельности и тратит годы на втягивание в ее ритм. Некоторые из них за три года так и не успевают привыкнуть к работе и активно включиться в нее и поэтому, когда представляется возможность, начинают искать что-то более для себя приемлемое (как им кажется).

Повысить реальную гибкость и эффективность системы высшего образования можно, по-видимому, отказавшись от единого срока обучения в вузе (около 5 лет) и от единого диплома о высшем образовании. Неоднократно выдвигавшиеся предложения о введении промежуточного этапа в высшем образовании после двух-трех лет обучения и соответствующего квалификационного документа заслуживают повторного и очень внимательного изучения. При этом следует предусмотреть возможность более или менее длительного перерыва между двумя этапами высшего образования, а также возможность перехода в другой вуз родственного профиля для завершения образования после первого этапа. Подготовка специалистов должна планироваться раздельно по этапам.

Часто высказывается мнение, что в связи с быстрым распространением сферы применения вычислительной техники, с вовлечением в эту сферу широких слоев работающего населения становится неизбежным и желательным использование ЭВМ без программирования. С этим можно согласиться, но только если иметь в виду программирование в его современном виде — составление более или менее объемистых программ на алгоритмических языках. Конечно, участвовать в написании прикладных программ, включаемых в пакеты и

...от программного обеспечения, в силу его большой гибкости, можно потребовать, чтобы оно автоматически настраивалось на индивидуальные психологические свойства пользователя, учитывало его уверенность, склонность к необдуманным действиям, степень доверия к машине и другие особенности личности, выявляло его типичные ошибки и способствовало их предотвращению.

фиксирующих накопленное алгоритмическое знание, должны немногие специалисты.

Но если применение вычислительной техники в какой-то области деятельности становится массовым, то все или почти все люди, причастные к этой деятельности, должны уметь поставить свою задачу. Минимальные требования к ним были сформулированы выше — они должны уметь прочесть и понять формальное описание модели предметной области и описание исследуемого объекта. Это уже предполагает известную культуру общения с вычислительной техникой. Более того, совершенно необходимо, чтобы спецификации задач, составляемые массовыми пользователями, были точными, исчерпывающими и не заставляли систему автоматического решения задач осуществлять лишние действия. Для этого авторы спецификаций должны достаточно глубоко понимать суть дела, понимать значение каждого элемента спецификации и знать, хотя бы в общих чертах, последствия решений, принимаемых при выборе варианта спецификации.

В этом смысле использование вычислительной техники без программирования оказывается невозможным.

В то же время системы автоматического решения задач, как и другое программное обеспечение ЭВМ, должны быть «дружественными» по отношению к пользователю, не обременять его лишними требованиями, а помогать ему в трудные моменты советами и разъяснениями.

Более того, от программного обеспечения, в силу его большой гибкости, можно потребовать, чтобы оно автоматически настраивалось на индивидуальные психологические свойства пользователя, учитывало его уверенность, склонность к необдуманным действиям, степень доверия к машине и другие особенности личности, выявляло его типичные ошибки и способствовало их предотвращению.

Резюмируя, можно выдвинуть на ближайшее будущее решение следующих проблем.

Факультативный подход к обучению студентов различных специальностей прикладной математике, программированию и другим методам использования вычислительной техники по принципу: минимум знаний — каждому, углубленная подготовка — всем заинтересованным.

Гуманизация программного обеспечения ЭВМ:

— разработка научных основ учета общечеловеческих факторов в архитектуре программного обеспечения и в языках общения человека с ЭВМ и ЭВМ с человеком;

— создание систем программного обеспечения, учитывающих индивидуальные особенности каждого пользователя и помогающих ему в общении с машиной.

8. Программирование как самостоятельная сфера деятельности

В настоящее время в центре научной проблематики большинства вычислительных центров и институтов прикладной математики и кибернетики, в том числе — академических, находятся математические методы решения конкретных классов задач, а вопросы программирования, перехода от задачи к программе отступают на задний план. Хотя это и является отражением объективно существующего приоритета задач перед программами, такое положение по ряду причин нельзя признать удовлетворительным.

В настоящее время в центре научной проблематики большинства вычислительных центров и институтов прикладной математики и кибернетики, в том числе — академических, находятся математические методы решения конкретных классов задач, . . . такое положение по ряду причин нельзя признать удовлетворительным.

Во-первых, основные классы задач являются массовыми, их приходится решать не только там, где разрабатываются методы решения и соответствующие им программы, но и во многих других местах. Это значит, что программы должны быть достаточно простыми в эксплуатации, надежными, легко модифицируемыми, хорошо документированными, должны обладать и другими потребительскими качествами. Иначе говоря, они должны представлять собой программный продукт. Изучение потребительских свойств программного продукта, разработка технологии производства программ, обладающих этими свойствами в должной мере, организация производства таких программ — все это вопросы, решение которых требует совсем иной квалификации, чем разработка математических методов решения задач.

Во-вторых, программы решения самых разных задач имеют много общего по своей внутренней структуре, организации, характеру функционирования. Это значит, что программирование порождает целый ряд понятий, методов и приемов работы, связанных именно с программами, как объектом деятельности, а не только как средством решения исходных задач. Программирование породило достаточно обширную и сложную научную проблематику, стало самостоятельной научной дисциплиной. Естественно, обнаружились тесные связи программирования с математикой (математической логикой, теорией графов, теорией вероятностей и другими разделами математики). Но программа — это все-таки не абстрактный, а вполне конкретный объект, поэтому теоретическое программирование не стало математической дисциплиной, а отпочковалось от математики, подобно тому, как это ранее произошло с механикой, теоретической физикой и т. п. Современное теоретическое программирование — это достаточно развитая и сложная область научной деятельности.

В-третьих, как всякая деятельность, программирование нуждается в своем инструментарии. Уже на заре программирования стали появляться трансляторы с алгоритмических языков различного уровня, редакторы текстов, системы отладки, кросс-системы и т. п. Все это — программы, помогающие создавать другие программы, они стали существенной частью системного программного обеспечения. К этому программному инструментарию добавились различные аппаратные средства — терминалы, рабочие места (станции) программиста, профессиональные персональные ЭВМ. Разработка программного и аппаратного инструментария для производства программ стала очень важной стороной деятельности программистов.

Все это означает, что программирование выросло до уровня обширной научно-производственной отрасли, нуждающейся в своей организационной структуре, своих институтах, производственных предприятиях, печатных органах. Более того, необходимо создать экономику этой отрасли, определить принципы ценообразования на программный продукт, разработать ряд нормативных документов. Нуждается эта отрасль и в надлежащей правовой основе, в частности, в решении вопросов авторского права на программы, а также ответственности за так называемые компьютерные преступления.

Назрел вопрос о выделении программирования в отдельную учебную специальность со специализациями: системное программирование, прикладное программирование, программирование для микроЭВМ. Необходимо разработать варианты учебного плана по этой специальности для университетов и технических вузов. Целесообразно также составить несколько типовых вариантов раздела учебного плана по методам использования ЭВМ (включая математические методы, программирование и программное обеспечение ЭВМ) для вузов различного профиля.

Статья поступила 18 июля 1984 г.

Эволюция элементной базы вычислительной техники: механические шестерни, электрические реле, электронные лампы, полупроводниковые триоды (транзисторы), интегральные схемы, микропроцессоры...
Что дальше: биокомпьютеры?

УДК 681.32:577.1

В. Л. Дунин-Барковский

ОЦЕНКИ ВОЗМОЖНОСТЕЙ БИОМОЛЕКУЛЯРНЫХ ВЫЧИСЛЕНИЙ

Жизнь живой клетки подобна работе сложной машины или даже целого производственного предприятия. В ней идут процессы получения энергии и «строительных материалов», из которых она состоит, процессы переработки и копирования информации. Детали этих процессов ни для одной клетки еще полностью не изучены, но уже сейчас можно количественно оценить уровень их сложности. Такие оценки с одной стороны дают возможность понять, насколько мы еще далеки от полного понимания механизмов «клеточной кухни», а с другой стороны они позволяют представить, какие потенциальные возможности открывает инженерное использование созданных природой *молекулярных машин* и их функциональных компонент. Ниже мы приведем оценки эффективности работы клеточных систем в терминах традиционной вычислительной техники и обсудим возможные варианты технического использования молекулярных биокомпонент. Реальных технических информационно-вычислительных устройств из биокомпонент в настоящее время еще нет. Поэтому в данной статье речь пойдет лишь о **потенциальных возможностях**: их обсуждение ведется сейчас в ведущих научных журналах Science [1], Nature [2], Proceedings of National Acad. Sci. [3], Вестник АН СССР [4].

Перейдем к оценкам — количественным и качественным. Первыми аналогию между логическими переключательными схемами и клеточной биохимией установили в 1961 году французские ученые лауреаты Нобелевской премии Жакоб и Моно [5]. Они рассмотрели возможности, связанные с тем, что продукты отдельных реакций могут блокировать или активировать синтез соответствующего белкового фермента-катализатора этой же или другой реакции.

Такая система может стать основой *логической вычислительной сети* достаточно большой сложности. Количественная оценка возможностей переработки информации подобными системами клетки может быть получена следующим образом [6]. В клетке животного насчитывается примерно 10^6 различных типов молекул, причем в течение одной секунды может появиться или исчезнуть до 10^3 молекул каждого типа [7]. Будем считать, что состояние клетки полностью определяется соотношением чисел молекул разных типов, т. е. влияние взаимного расположения молекул и других подобных факторов на состояние клетки учитывать не будем. Полагаем, что число молекул каждого вида изменяется в интервале 0—1000. Таким образом состояние клетки может быть задано миллионом 10-разрядных *двоичных* векторов. Другими словами, емкость «оперативной памяти» клетки имеет порядок 10^7 бит. Любой из разрядов, описывающих состояние клетки, может измениться в течение одной секунды 10^9 раз. Эти изменения соответствуют *«информационной производительности»* около 10^9 бит/с. Если подобным образом оценить информационную производительность микроЭВМ «Электроника 60», исходя из быстродействия 200 тыс. 16-битовых операций/с и оперативной памяти 32 К, то получим $6 \cdot 10^6$ бит/с. Иными словами, использование вычислительных возможностей биохимических реакций может позволить иметь в объеме одной клетки (около 1000 мкм^3) вычислительное устройство с быстродействием и оперативной памятью большими, чем у современных малых ЭВМ.

Одну конкретную систему элементов-молекул в качестве базы для вычислений в клетке рассмотрели советский биофизик Е. А. Либерман [8] и американский физик И. Беннет [9]. Они обратили внимание на то, что копирование молекул ДНК при делении клеток — это по-существу операция, аналогичная дублированию цифровой информации (например, перезаписи с диска на диск для другого пользователя — одной из наиболее распространенных в вычислительных системах операций. Для технических приложений биологических систем важно знать, что копирование ДНК клетки — «ленты», содержащей около 10^9 бит информации, может завершиться примерно за один час, т. е. скорость перезаписи имеет порядок 10^6 бит/с — величину, обычную для существующих технических систем. При этом вероятности ошибок таковы, что не более 1% копий («потомков») получаются дефектными. Вероятность ошибки при воспроизведении 1 бита составляет 10^{-11} и относится к техни-

чески приемлемым величинам вероятности ошибки.

Воспроизведение — лишь одна операция из целого *комплекса операций,* связанных с ДНК в клетке, известного под названием «процессинг ДНК и РНК» [8, 10]. При процессинге происходит разрезание ДНК на фрагменты и последующая их «сшивка» в тех или иных местах. **Советские биофизики М. Н. Вайнцвайг и Е. А. Либерман показали, что система операций указанного типа полна и с их помощью можно моделировать универсальную машину Тьюринга** [11]. Иными словами *возможно вычислительное устройство, основные операции которого заключаются в манипулировании «словами», состоящими из отрезков ДНК.* Подобное манипулирование словами может служить реализацией универсального алгоритмического языка, кстати, весьма похожего на широко известный списковый язык ЛИСП.

Для получения некоторой обобщенной оценки эффективности различных вычислительных конструкций Е. А. Либерман ввел понятие «цены действия» — произведение величины *энергии,* затрачиваемой на одну бинарную логическую операцию, на *время* выполнения этой операции *. Для биомолекулярных вычислений элементарная операция (разрыв или «сшивка» цепи ДНК) — это *разрыв* или *образование* одной ковалентной связи**. Энергия, необходимая для этой операции, имеет порядок 1 эВ ($1{,}6\cdot10^{-19}$ Дж), а время выполнения действия — 10^{-3} с. Таким образом *цена действия* вычислительного устройства, использующего в качестве элементарных операций разрыв и формирование ковалентных связей, равна $1{,}6\cdot10^{-22}$ Дж·с$\sim$ $\sim10^{12}\hbar$.

Для сравнения отметим, что в микроЭВМ «Электроника 60» при потребляемой мощности 130 Вт и указанном выше эквивалентном «битовом» быстродействии $6\cdot10^{6}$ бит/с, цена действия равна:

$$130\ \text{Дж/с}\cdot(6\cdot10^{6})^{-2}\text{c}^{2}=3\cdot10^{-12}\ \text{Дж}\cdot\text{с}\sim 3\cdot10^{22}\hbar.$$

Иными словами, промышленно выпускаемые ЭВМ уступают по цене действия молекулярным вычислителям на много порядков.

В организме человека 10^{12} клеток. Каждая из них, как сказано выше, потенциально может быть компьютером средней мощности. Каковы потенциальные возможности сети из 10^{12} таких машин, которая могла бы существовать в организме? Нетрудно убедиться, что в большинстве тканей организма количество не может перейти в качество. В самом деле, клетки большинства органов связаны диффузионно или потоками крови и лимфы. Скорость этих потоков невелика — около 10 см/с. На расстояниях менее 200—300 мкм из-за вязкости течения жидкостей любые вещества могут распространяться только посредством диффузии. Среднее время прохождения любой молекулой расстояния 300 мкм в водном растворе — 90 с. Таким образом, время обмена любой информацией между клетками не менее одной минуты. Это означает, что сети из астрономического числа клеточных компьютеров не могут быть особенно эффективны в условиях организма.

Существенными преимуществами в отношении обработки информации по отношению к другим тканям организма обладает нервная. Здесь диффузия ограничена межнейронным пространством — синаптической щелью или синапсом, имеющим размеры не более 10^{-5} см. На большие расстояния информация распространяется по нервным волокнам с помощью автоволновых механизмов со скоростью 1—100 м/с. В связи с этим сделано предположение о том, что наблюдаемые в физиологических экспериментах нервные импульсы — это обмен информацией между чрезвычайно эффективными молекулярными компьютерами — отдельными нервными клетками. Иными словами, выдвигается гипотеза о том, что основную свою работу нервная система выполняет на *биомолекулярном,* а *не импульсном* уровне [8, 12]. Впрочем нет необходимости объяснять высокую эффективность мозга, следуя именно этой гипотезе. Даже если оценивать работу мозга, исходя из классических представлений о нем, как о схеме пороговых элементов с изменяющимися связями, его эффективность и операционные характеристики оказываются очень высокими. В самом деле, в мозгу человека около 10^{11} нейронов, каждый из которых имеет связи с 1000 другими нейронами через синапсы [6]. Можно считать, что в каждом синапсе реализуется бинарная логическая функция. «Такт» работы нервной системы — 1 мс. Таким образом, в мозгу в целом может совершаться до 10^{17} бинарных операций/с. На эту работу человек затрачивает не более 10^{7} калорий в сутки или 100 калорий/с. Нетрудно убедиться, что в этом случае цена действия в нервной системе в целом составляет $5\cdot10^{-32}$ Дж·с$=500\,\hbar$. Иными словами, удельная эффективность и полная производительность современных средних по мощности средств вычислительной техники

---

* Любопытно, что размерность цены действия совпадает с размерностью постоянной Планка — $\hbar$ ($\hbar=1{,}06\cdot10^{-34}$ Дж·с).

** Тип химической связи, при котором объединяются «электронные облака» связанных атомов.

(см. выше оценки для микроЭВМ «Электроника 60») в 10^{20} (!) раз уступают возможностям мозга.

Количественные оценки эффективности биомолекулярных конструкций являются мощным стимулом для попыток найти пути для создания инженерных информационно-вычислительных систем на биологической и биомолекулярной основе. Опишем кратко некоторые из предложенных в последнее время и, возможно, разрабатываемых сейчас проектов такого рода. Следует сразу оговориться, что техническая реализация этих проектов возможна лишь в отдаленном будущем (1990—2000 годы).

Первый из обсуждаемых проектов — проект непосредственного ввода огромных массивов информации в мозг. Он разрабатывается как проект зрительного протеза [1]. Предполагается создать матрицу из 10^6 фотоэлементов, размером с естественную сетчатку глаза и поместить ее в физический аналог глаза. Потенциалы, генерируемые фотоэлементами при попадании на них света, должны восприниматься чувствительными отростками дендритов нейронов. Нейроны, в свою очередь, должны «вселяться в мозг» с помощью активно разрабатываемых сейчас методов имплантации нервных клеток, выращенных в условиях культуры нервной ткани [13]. Нейроны будут превращать световые сигналы в нервные импульсы и будут предприняты меры, чтобы аксоны этих нейронов прорастали непосредственно в мозг в зону, соответствующую естественной зрительной коре.

Интересным является предложение использовать ДНК (или РНК) в качестве ленты для хранения информации [1]. Проект состоит в том, что произвольная информация (печатные тексты, графика, звуковые сигналы и т. д.) кодируется фиксированным образом в последовательность нуклеотидов молекулы ДНК, которая таким образом синтезируется в устройстве, осуществляющем кодирование. В декодирующем устройстве молекулы ДНК преобразуются в те сигналы, которые в них закодированы. Стимулом для использования «информационного полимера» типа ДНК для хранения произвольной информации является высокая объемная плотность записи информации: **10^{20} бит на один грамм записывающего вещества.** Использование для этой цели именно ДНК оправдано тем, что эта молекула уже продемонстрировала свою способность хранить информацию, а ее компоненты настолько распространены в природе, что могут считаться относительно дешевыми (у каждого человека имеется около 10 г ДНК). Чтобы представить эффективность хранения отметим, что **в 1 г вещества ДНК-памяти может быть записано содержание всех книг мира.** В самом деле — большая книга — это 10^6 бит информации. Если к тому времени, когда на Земле будет проживать 10 млрд. человек, на каждого из них будет приходиться по 10^3 книг, то вся информация в этих книгах составит 10^{19} бит, т. е. меньше, чем информация в одном грамме ДНК. К сожалению, известные в настоящее время методы химического синтеза ДНК в лабораторных условиях* позволяют наращивать длину ДНК со скоростью лишь два нуклеотида в час (т. е. записывать на ДНК информацию со скоростью 4 бит/с). Примерно такого же порядка и достижимая сейчас скорость расшифровки структуры ДНК [14], т. е. скорость считывания информации. В этих условиях запись даже одной книги займет 30 лет.

Вместе с тем, в естественных условиях, как уже отмечалось выше, синтез ДНК может идти со скоростью 1000 нуклеотидов/с, т. е. в миллионы раз быстрее. При этом одновременно могут синтезироваться различные молекулы ДНК или различные фрагменты одной молекулы. Разумеется, этот процесс синтеза идет в работающих живых клетках и синтезируются те молекулы ДНК, которые нужны этим клеткам. Для того, чтобы воспользоваться «технологией синтеза», применяемой живыми клетками, нужно научиться *управлять программами* работы клеток, т. е. создать из них как бы *микроскопические биохимические работы.* Эта идея находится сейчас в стадии разработки постановки проблемы [3].

В качестве начального этапа решения подобных задач рассматривается проблема инженерного конструирования белков с заданными свойствами [2]. Первые примитивные достижения в этом направлении уже есть, но вся основная работа еще впереди [15]. Легко понять, почему это так: для того, чтобы было возможно *инженерное конструирование макромолекул и систем, подобных живым клеткам,* необходимы годы многоэтапных исследований.

Еще одной, пока далекой от практической реализации проблемой, является *проектирование и создание молекулярных компьютеров* с размерами и свойствами, подобными живым клеткам. Речь идет о том, чтобы используя потенциальные вычислительные возможности, которые имеются в клетке, создать технически реализуемые цифровые вычислительные устройства. Как уже говорилось, потенциальные возможности позволяют в клетке

* Этими методами, например, был синтезирован искусственный ген [14].

объемом 10^3 мкм^3 иметь микропроцессор весьма высокого уровня. В 1 см^3 может поместиться 10^6 таких машин. Любопытно отметить, что один из способов, которым предлагается решить проблему энергообеспечения и термостабилизации этих компьютеров, состоит в их подкожной имплантации человеку [1]. В этом случае возникает принципиальная возможность сопряжения встроенных компьютеров с работой мозга непосредственно внутри человеческого организма, минуя традиционные средства ввода-вывода (клавиатуру, дисплей, речевой ввод-вывод). Иными словами, имеется возможность «мысленного» управления работой этих вычислительных устройств и получения результатов непосредственно нервной системой.

На этой ноте, пока звучащей видимо фантастически, имеет смысл закончить обзор потенциальных возможностей молекулярных вычислений.

Стоит лишь отметить, что в ноябре 1983 года в Сан-Франциско под председательством М. Конрада (Дейтройтский университет) состоялось первое рабочее совещание по проблеме молекулярных вычислений. М. Конрад в 1980 году и в 1982—83 годах работал в Москве в институте проблем передачи информации АН СССР и подготовил несколько совместных публикаций с Е. А. Либерманом и его сотрудниками. В работе совещания участвовало около 30 человек — специалистов по вычислительной технике, прикладной математике, физике, нейрофизиологии, молекулярной биологии и электронике (в их числе — физик, лауреат Нобелевской премии Р. Фейнман, известный нейрофизиолог Р. Ллинас и другие). Судя по интервью М. Конрада в широкой прессе, на начальные работы в данной области выделяются значительные средства.

ЛИТЕРАТУРА

1. R o b i n s o n A. L. Nanocomputers from organic Molecules. — Science, 1983, vol. 220, No 4600, p. 940—942.
2. P a b o C. Designing proteins and peptids. — Nature, 1983, vol. 301, No 5897, p. 200.
3. D r e x l e r K. E. Molecular engineering: An approach to the development of general capabilities for molecular manipulation. — Prac. Nat. Acad. Sci. USA, 1981, vol. 78, No 9, p. 5275—5278.
4. И в а н и ц к и й Г. Р. Биологические микроустройства. — Вестник АН СССР, 1984, № 3, с. 118—128.,
5. J a c o b F., M o n o d J. Genetic regulatory mechanisms in the syntethesis of proteins. — Journal of Molecular Biology, 1961, vol. 3, No 3, p. 318—356.
6. Д у н и н-Б а р к о в с к и й В. Л. Информационные процессы в нейронных структурах. — М.: Наука, 1978, — 166 с.
7. У о т с о н Д ж. Молекулярная биология гена. — М.: Мир, 1976. — 400 с.
8. Л и б е р м а н Е. А. Живая клетка. — М.: Наука, 1982. — 143 с.
9. B e n n e t Ch. H. The thermodynamics of computation — a review. — Int. J. Theoret. Phys., 1982, vol. 21, No 12, p. 905—940.
10. P e r r y R. P. Processing of RNA: — Ann. Rev. Biochem., 1976, vol. 45, p. 605—625.
11. В а й н ц в а й г М. Н., Л и б е р м а н Е. А. Формальное описание молекулярной вычислительной машины (МВМ). — Биофизика, 1973, т. 18, № 5, с. 939—942.
12. C o n r a d M. Microscopic-macroscopic interface in biological information processing. — Detroit: Wayne State University, Feb., 1983 (preprint). — 51 p.
13. П о л е ж а е в Л. В. Трансплантация ткани мозга и некоторые проблемы биологии и медицины. — Вестник АН СССР, 1984, № 6, с. 23—33.
14. Genetic Engineering. Principles and Methods.N.-Y., 1983, vol. 5, 1983.
15. R i c h a r d s o n J. S. De novo desing and synthesis of a protein. — B i o p h y s. J., 1984, vol. 45, No 2 (part 2), p. 16.

Статья поступила 10 июля 1984 г.

УДК 681.32:577.155.3

Н. Н. Всеволодов

ПРИМЕНЕНИЕ „БИОХРОМА" ДЛЯ СОЗДАНИЯ БИОКОМПЬЮТЕРА

Биологические микроустройства и фотобиоэлементы

Попытки воспроизвести в технике принципы работы биологических (живых) систем предпринимались в глубокой древности. В настоящее время это раздел науки — бионика. Сегодня создаются гибридные системы из технических и биологических элементов-деталей. Начаты работы по созданию чисто биологических систем технического назначения. Что при этом в первую очередь привлекает конструкторов? Высокий (близкий к 100%) КПД, миниатюрные размеры (как правило, более, чем на три порядка меньше размеров современных «небиологических» конструкций того же назначения), высокая чувствительность, экологическая совместимость и широкий спектр применения. Интерес к так называемым биологическим микроустройствам (клеточным и молекулярным комплексам) стремительно растет [1].

Слово *биокомпьютер* из фантастической литературы в научную перекочевало сравнительно недавно [2]. В промышленно развитых странах начаты работы по созданию биокомпьютера (биологической вычислительной машины) на б и о л о г и ч е с к и х м и к р о у с т р о й с т в а х (биологических датчиках элементов нервной системы); исполнительных и выводных устройств на основе биологических клеточных и молекулярных комплексов [3].

Бум в исследовании биологических мембран начался 20 лет назад. Доказано, что

биологические мембраны — это биологические «печатные платы» со «впаянными» в них биоэлементами (молекулами и молекулярными комплексами). Взаимодействуя между собой биоэлементы обрабатывают и транслируют не только вещества, но и информацию [4]. В отличие от традиционных элементов микроэлектроники, биоэлементы могут вращаться и перемещаться в плате-мембране, так как она представляет собой полужидкий липид-белковый слой. Биоэлементы взаимодействуют на физическом, химическом и биохимическом уровнях. В первом случае времена взаимодействия могут быть пико- и наносекунды, в последнем — минуты и часы.

Широкий диапазон качественных и временных взаимодействий позволяет системе (например, клетке или организму) оптимально взаимодействовать с окружающей средой.

Проникновение физических методов исследования в биологию дало большое количество модельных систем на основе искусственных и естественных мембран, в которых моделируются практически все процессы в биомембранах и клетках. Наряду с генной инженерией появилась клеточная инженерия.

Биотехнология — это не только направление науки, но и промышленность. Ее достижения позволяют уже сейчас создавать биосистемы и биоустройства с наперед заданными свойствами и программой работы (измененный «биоинженерами» микроб, превращающий отходы химии в белок, можно назвать биоустройством).

Идея **биокомпьютера**, т. е. биологической вычислительной системы или машины с заданной программой работы сегодня имеет вполне реальную материальную и научную базу.

Обработка информации в биологических системах

Скорость передачи сигналов между «блоками» ввода, обработки и вывода информации в биосистемах, как полагают, намного ниже, чем в блоках современных ЭВМ. Однако уникальная способность мозга к параллельной и ассоциативной обработке информации делает человеческое мышление несопоставимо более эффективным. Объем памяти мозга необычайно высок: по некоторым оценкам человек к 80 годам должен иметь 10^{20} бит информации об окружающей среде. Почти вся эта информация вводится в форме видеообразов с огромной скоростью, определяемой наличием в сетчатке глаза множества независимых параллельных каналов. Это заставляет конструкторов перспективных ЭВМ уделять столько внимания оптическим системам обработки информации [5]:

Таблица

Развитие средств обработки информации

Устройства обработки информации	Способ передачи и обработки сигналов	Годы
Механические	Механический	
Ламповые	Электронный	1950-е
Полупроводниковые	»	1960-е
Интегральные схемы	»	1970-е
Оптические	Световой	1980-е
Биологические	Электронный, световой, транспорт ионов, биохимические реакции и..?	1990-е

Ввод и вывод данных в ЭВМ будущих поколений должен быть в форме, привычной для человека, — в виде звука и образа. Мозг обрабатывает видеосигналы параллельно из всех точек сетчатки глаза. Считается, что еще до попадания в мозг часть информации обрабатывается так, чтобы в мозг шел не весь массив данных, а только требуемый для решения конкретной задачи, наиболее важной в данный момент, т. е. происходит распределенная обработка-фильтрация информации.

Фотохромные материалы для обработки информации

К настоящему времени созданы многие элементы интегральной оптики, системы ввода и вывода видеоинформации.

Запись в долговременную оптическую память производится на оптические диски, покрытые фоточувствительным слоем с помощью сфокусированного лазерного луча. Оперативная информация обрабатывается в фоточувствительных реверсионных средах — фотохромах, на которых сразу после экспозиции появляется изображение, которое перед новой записью может быть стерто.

В оптических решающих устройствах обработка (преобразование) изображения из 500×500 точек занимает несколько миллисекунд (у ЭВМ традиционной архитектуры — десятки минут). Обработка в оптической системе происходит сразу по всей площади амплитудным или фазовым (по изменению коэффициента преломления) способом.

Фазовый способ записи видеоинформации используется в голографических системах и отличается от амплитудного большой избыточностью информации — каждая деталь изображения «размазана» по всей площади среды, и повреждение какого-либо участка не искажает информацию (на оптических же дисках легкая царапина может привести к выпадению нескольких битов информации).

Впрочем, отметим, что у обоих способов записи есть и достоинства и недостатки [6].

Появление голографии и лазеров привело к созданию новых способов регистрации и обработки информации. Это, в свою очередь, потребовало разработки новых фоточувствительных сред, таких, как фотохромы [7]. Именно они — важнейший элемент оптических вычислительных машин (ОВМ), подобно магнитным лентам или дискам в ЭВМ. Разработаны фотохромы различных типов на неорганических и органических материалах. От них требуются высокая разрешающая способность (для голографических систем памяти не менее 10 000 лин./мм); большая цикличность «запись—стирание—запись» (для надежной работы ОВМ не менее 100 000); чувствительность не менее $5 \cdot 10^{-4}$ Дж/см^2 (это специальный параметр чувствительности фотохромов — при такой энергии света поглощение фотохрома изменяется не менее, чем на 0,2 опт. ед.); дифракционная эффективность достигает нескольких процентов (этот параметр определяет чувствительность фотохрома к голографическому способу записи). При голографическом способе на одну пленку толщиной 1 мм можно записать до 100 голограмм, не смешивающихся при считывании. Это резко увеличивает плотность записи. В последнее время в нашей стране создан новый способ — поляризационно-голографический [8]. Он позволяет увеличить количество голограмм на одной пленке на порядок.

Из сказанного видно, что необходимый для решающей оптики фоточувствительный материал должен обладать уникальными свойствами — регистрировать все четыре параметра светового луча: амплитуду, фазу, поляризацию и (желательно) длину волны, т. е. цвет. Пока нет еще фотоматериала, обладающего полным комплексом перечисленных характеристик, но дело в том, что до сих пор поиск шел только среди неорганических и органических (синтетических) материалов. Биологические упоминались лишь как экзотические, не имеющие практического значения.

Хлорофилл и его аналоги используются растениями и фотосинтезирующими организмами, родопсин — животными для ориентации и регистрации видеообразов. Практически кроме родопсинов (ретиналь-белковых комплексов) и хлорофилл-белковых комплексов на планете не обнаружено других светотрансформирующих биосистем.

Процесс зрения напоминает фотохромный, только для восстановления засвеченного в том или ином участке сетчатки родопсина организм использует специальный фермент.

Попытки использовать биологические вещества и целые молекулярные комплексы для регистрации изображения изредка делались.

Если прикрыть лист растения какой-либо фигурой, вырезанной из непрозрачной бумаги, то на затененной площади синтез хлорофилла прекращается, цвет изменяется и получается негативное изображение. Более того, если из листьев выращенного в темноте растения выделить предшественник хлорофилла (протохлорофилл), и сделать из него пленку, то он, превращаясь под светом в хлорофилл, изменяет цвет, — получается изображение.

На фотографические пленки из биологического материала уже имеются патенты [9, 10]. Однако эти биофотоматериалы нестабильны, а приготовить из них пленки — сложно и дорого.

Элементы биокомпьютера из «биохромных» пленок

Ситуация с биофотоматериалами резко изменилась 10 лет назад, когда был выделен аналог родопсина — бактериородопсин [11]. В отличие от зрительного родопсина после экспозиции бактериородопсин самостоятельно (без действия фермента) возвращается в исходное состояние (т. е. это оказался чисто фотохромный материал).

Молекулы бактериородопсина в виде одномерной гексагональной решетки занимают в мембране бактерии участок примерно в 1 мкм в поперечнике (толщина участка 4—5 нм). Упаковка в виде двумерной гексагональной решетки придает необычайную устойчивость молекулам бактериородопсина. После выделения из клетки этот участок сохраняет свои свойства. В нем находится 100—200 тыс. молекул бактериородопсина. Такие фрагменты мембран названы пурпурными мебранами (ПМ) и именно с ними работают биологи и биофизики.

Из пурпурных мембран в 1978 г. была сделана первая фотохромная пленка*, названная «биохромной», которая благополучно работает и сейчас. Это говорит об устойчивости пленок на основе ПМ к внешним воздействиям (хранение было не специальное). С помощью метода биохимической реконструкции делаются и цветные пленки. Технологии получения пурпурных мембран и пленок из них просты, безвредны, экологически чисты.

Выяснено, что «биохромная» пленка обладает рекордной цикличностью (более 100 000) и чувствительностью, предельной для фотохромов. Разрешающая способность — выше 10 000 лин./мм, так как размер светочувстви-

* Работа выполняется в рамках проекта «Родопсин». Руководитель — академик Ю. А. Овчинников.

тельного элемента — молекулы бактериородопсина — 3—4 нм [12].

Пленка из «биохрома» чувствительна ко всем параметрам луча света. Это больше удивляет физиков, чем биологов, так как фоторецепторы животных и бактерий должны обладать универсальными свойствами, чтобы выполнять различные функции в глазах рыб, насекомых, животных, живущих в разных средах и имеющих разные способы зрения и обработки изображения. «Биохром» способен к поляризационной амплитудной и голографической записи.

Вернемся к «биохрому». Скорость записи на «биохроме» может достигать пикосекунд. Биохимической и химической модификацией можно изменять время хранения записанной информации от долей секунды до нескольких недель. Поэтому можно обрабатывать информацию в реальном масштабе времени с наносекундным разрешением во времени.

Дифракционная эффективность уже сейчас несколько процентов. Недавно получена запись первой голограммы. Длительное освещение в полосу поглощения лазерным светом мощностью до 100 Вт/см^2 не приводит к необратимым изменениям.

Можно получать монослои 5 нм и слои толщиной миллиметр. Ведутся работы по получению кристаллов бактериородопсина и на их основе фотодиодов с пикосекундным разрешением. Есть принципиальная возможность сделать пленку с постоянной фиксацией изображения и с усилением, т. е. достигнуть фотографических чувствительностей. Работы практически только начаты, и что еще «может» биохром, покажет ближайшее будущее.

Перечислим, что можно предложить для автоматизированной обработки информации в будущем биокомпьютере, если использовать известные на сегодня свойства «биохрома».

1. Постоянная оптическая память. Мощный ультрафиолетовый свет необратимо изменяет цвет пленки. При записи с разрешением 500—1000 лин./мм на 1 см^2 можно поместить до 10^8 бит информации, а в стопке из монослоев ПМ толщиной 1 см (т. е. в 1 см^3) — до 10^{14} бит.

2. Постоянная голографическая и оперативная память. Первая достигается записью ультрафиолетовым, вторая — видимым светом. Плотность голографической записи — обычная (в пленке до 100 голограмм). Но так как «биохром» обладает и поляризационной чувствительностью, то общая емкость повышается по отношению к традиционной фазовой еще на порядок.

3. Оптический переключатель с временами действия от пикосекунд до минут в зависимости от интенсивности света и длины волны.

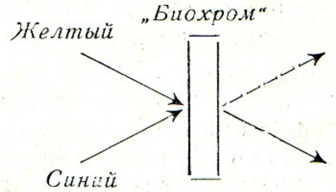


Рис. 1. Обратимый переключатель. При большой интенсивности синего света блокируется проход желтого и наоборот. Введение поляроидов позволяет модулировать и переключать лучи внешним (поляризованным определенным образом) лучом

Можно сделать переключатель, срабатывающий на какой-либо избранной длине волны при комбинации двух-трех других длин волн (рис. 1).

4. То же, но с использованием поляризационных свойств.

5. Микрофиши и биомикросхемы с разрешением до 0,01 мкм. Показано, что электронный луч в вакууме необратимо выжигает фоточувствительные центры «биохрома». Это дает возможность в будущем создавать биомикросхемы для биокомпьютера так же, как БИС для ЭВМ, но существенно большей плотности.

Обратимое изменение спектральных свойств ПМ под действием электрического и магнитного полей, pH, давления, температуры, химических реагентов и т. д. возможно также найдет применение в будущих системах обработки информации.

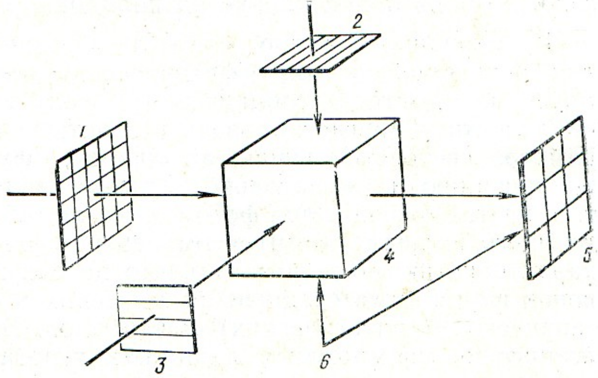


Рис. 2. Возможная система обработки оптической информации на элементах из материала «биохром»:
1 — исходная информация, записанная на пленке «биохром» ультрафиолетовым светом или электронным лучом, обычным или голографическим способом; *2, 3* — управляющие элементы на основе постоянных или оперативных элементов из серии «биохром»; *4* — элемент для обработки оптической информации на быстродействующем элементе из «биохрома»; *5* — экран вывода и наблюдения с длительным временем хранения оптической информации (в виде изображения) на «биохроме»; *6* — лучи стирания

Способ может быть поляризационным, голографическим, обычным или комбинацией этих способов. Информацию можно обрабатывать красным светом, стирать ее — синим (для обычного «биохрома») или комбинациями цветов (для аналогов «биохрома»).

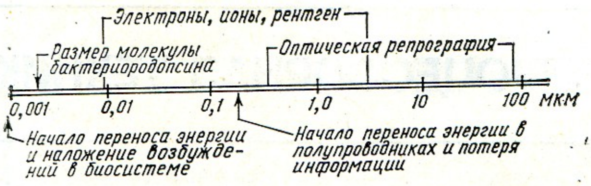


Рис. 3. Разрешающая способность излучений, применяемых для получения микрофиш, микросхем и т. д. Расстояния между микроэлементами, при которых начинается перенос энергии

В настоящее время можно вообразить схему на «биохромных» элементах для биологической вычислительной машины будущего (рис. 2). Возможность использования биологических микроустройств в технических системах уже рассматривается [1]. Однако «биохром» — лишь первая капля из того потока, который должен в ближайшее время хлынуть из лабораторий биологов в технологию.

Еще одно важное замечание: сейчас 100 000 полупроводниковых элементов можно расположить на 1 мм². При увеличении их числа расстояние между ними становится уже меньше микрона. Это приводит к потере информации из-за туннелирования электронов между полупроводниковыми элементами. Биомолекулярные элементы в тысячу раз меньше, однако в силу специфических свойств их паразитное взаимодействие отсутствует (или управляемо) даже при расстоянии между ними 10^{-3} мкм (рис. 3).

Уплотнение микроэлементов в ЭВМ приводит к весьма трудноразрешимой проблеме теплоотвода, биокомпьютер же на 50—80% состоит из воды. Это уже существенно облегчает решение проблемы теплоотвода. Вместе с тем высокий КПД биосистем резко уменьшает часть энергии, отводимой в тепло.

Энергетическое обеспечение биокомпьютера может быть самое различное — от электроэнергии и энергии света — до молекулярных источников типа АТФ*. Например, можно использовать систему на основе хлорофилла или родопсина для преобразования солнечной энергии в запасы АТФ для постепенного расходования последних. Идеалом, возможно, будет атомная батарейка, но заманчиво построить компьютер, полностью биологический (вплоть до энергетического источника).

ЛИТЕРАТУРА

1. Иваницкий Г. Р. Биологические микроустройства. — Вестник АН СССР, 1984, № 3, с. 118.
2. Tucker J. B. Biochips: can molecules compute? — High Technology, 1984, Feb., p. 36—47.
3. Yanchinski S. And now — the biochip. — New Scientist, 1982, N 11, p. 68—71.
4. Либерман Е. А. Молекулярная вычислительная машина. — Биофизика, 1978, № 6, с. 1118—1121.
5. Оптическая обработка информации. — М.: Мир, 1980. — 460 с.
6. Антонов Е. А., Гинсбург В. М., Лекциер Е. Н. Оптическая голография. Практическое применение. — М.: Сов. радио, 1978. — 238 с.
7. Барачевский В. А., Лашков Г. И., Цехомский В. А. Фотохромизм и его применение. — М.: Химия, 1977. — 280 с.
8. Какичашвили Ш. А. Поляризационная голография. — Вестник АН СССР, 1982, № 7, с. 51.
9. Pat. No 1423992 (Англ.). Photosensitive material of biological origin.
10. Pat. No 4084969 (USA). Photographic elements containing vesicles of rhodopsin and lipids.
11. Oesterhelt W., Stoeckenius D. Functions of a new photoreceptor membrane. — Proc. Nat. Amer. Sci., 1973, 70, p. 2853.
12. Всеволодов Н. Н. Биологические комплексы как фоторегистрирующие материалы. — IV Всесоюзная конференция «Бессеребряные и необычные фотографические процессы. Суздаль, 28.II—2.III 1984 г.». — Черноголовка», 1984, с. 118.

* АТФ — аденозинтрифосфат — органическое соединение, накапливающее, сохраняющее и транспортирующее энергию для обмена веществ организмов.

Статья поступила 10 июля 1984 г.

ИНФОРМАЦИЯ

Международный центр научной и технической информации подготавливает в серии «Методические материалы и документация по пакетам прикладных программ» специализированный выпуск «Локальные вычислительные сети: опыт международной стандартизации».

Данный выпуск сборника посвящен вопросам международной стандартизации локальных вычислительных сетей типа «ETHERNET». Рассматривается состояние работ Европейской ассоциации производителей ЭВМ (ЕСМА), американского Института инженеров по радиоэлектронике и электротехнике (IEEE), Международной организации по стандартизации ISO в сфере стандартизации функций нижних уровней локальных вычислительных сетей, включая транспортный уровень.

Для системы коаксиального кабеля, образующей общую среду связи сети, указываются требования к конструкции и электрическим характеристикам собственно кабеля, а также основные требования к соединителям, повторителям и способам подсоединения станций к общей среде.

Для функций физического уровня указываются требования к набору услуг, временным и электрическим характеристикам сигналов на международных интерфейсах.

Для функций канального уровня определяется структура кадра, рекомендуемые значения специфических параметров и протокол случайного доступа с контролем несущей и обнаружением конфликтов.

Стандартизованный протокол ЕСМА для транспортного уровня применяется как в локальных, так и в «больших» сетях с коммутацией пакетов и предусматривает средства установления и ликвидации транспортного соединения, обнаружения и исправления ошибок типа потери и дублирования блоков данных, регулирования потока и обнаружения пассивного состояния удаленного партнера.

Ориентировочная цена выпуска — 1 руб. 80 коп.

Заказы на выпуск принимаются магазином № 93 «Книга—почтой» Москниги по адресу: 117168, Москва, В-168, ул. Кржижановского, д. 14, корп. 1.

УДК 681.3.01:51

С. Т. Хвощ, В. Б. Смолов, А. И. Сухопаров, В. В. Горовой, Д. Н. Черняковский, Б. Н. Чернуха, А. Н. Васильев

КОМПЛЕКТ БИС ДЛЯ ОРГАНИЗАЦИИ МУЛЬТИПЛЕКСНЫХ КАНАЛОВ МЕЖМОДУЛЬНОГО ОБМЕНА ИНФОРМАЦИЕЙ

Комплект интерфейсных микропроцессорных БИС серии К588 позволяет разработчикам систем сбора, обработки и управления объектами в реальном масштабе времени организовать мультиплексные каналы обмена данными между микроЭВМ и удаленными периферийными устройствами в коде «Манчестер-II».

Обобщенная схема блока последовательного интерфейса, связывающего мультиплексный канал и общую шину микроЭВМ «Электроника 60М», приведена на рис. 1. Входной и выходной каскады служат для электрического согласования входов и выходов кодера-декодера манчестерского кода [1, 2] с трансформатором гальванической развязки мультиплексного канала. Селектор адреса канала сравнивает зашитый на его входах 5-разрядный адрес (ADRM) с кодом адреса в командах, приходящих из мультиплексного канала. В случае их совпадения или прихода адреса 11111 (общий режим), вырабатывается строб приема сообщения. Преобразователь последовательно-параллельных кодов служит для стыковки униполярных последовательных входов-выходов кодера-декодера с параллельной 16-разрядной шиной данных-адреса. Селектор адреса общей шины компарацией 12-разрядного кода, зашитого на входах ADRK с адресом в общей шине, различает команды ввода-вывода, относящиеся к блоку последовательного интерфейса. Контроллер общей шины и магистральные приемопередатчики согласуют уровни электрических сигналов и реализуют протокол обмена информацией по ОСТ 11.305.903—80.

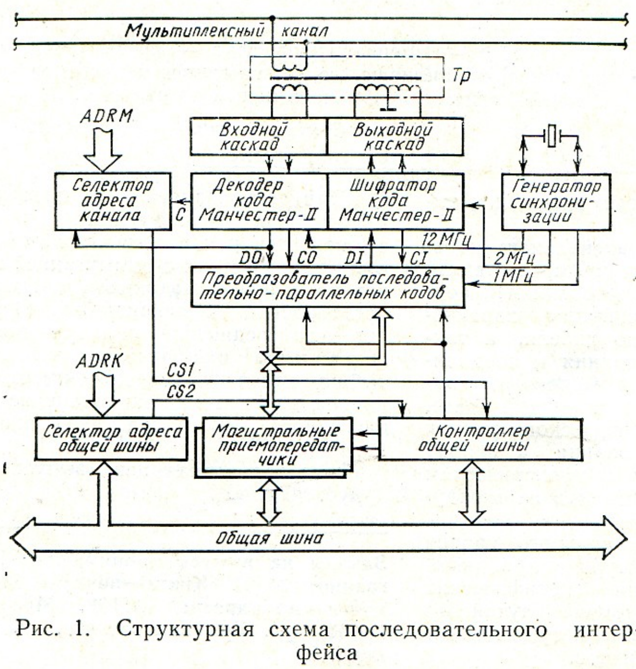


Рис. 1. Структурная схема последовательного интерфейса

Для эффективной организации мультиплексных каналов применяется комплекс БИС, в состав которого входят [3, 4]: формирователь трансформаторной линии (ФТЛ); усилитель-ограничитель (УО); кодер-декодер манчестерского кода (КДК); адаптер канала (АК); селектор адреса (СА); магистральный приемопередатчик (МПП) (табл. 1). Все схемы работоспособны в диапазоне температур —60 . . . +125° С.

Бис ФТЛ (рис. 2, *а*; табл. 2). ФТЛ усиливает электрические сигналы при передаче и преобразует биполярный код в униполярный фазоманипулированный при приеме данных. В составе БИС входят передатчик, приемник и схема управления для выработки сигнала начального сброса и управления направлением передачи данных при ретрансляции сигналов.

На рис. 2, *б* приведена схема подключения параллельного (БИС АК) или последовательного униполярного (БИС КДК) канала к мультиплексному каналу с использованием импульсного линейного трансформатора ТИЛ-1В для гальванической развязки с чис-

Таблица 1

Основные системотехнические параметры БИС

Обозначение БИС	Назначение	Дополнительные функции	Разрядность, бит: информационного слова	Разрядность, бит: шины данных	Ток потребления, мА	Напряжение питания, В	Технология изготовления	Число выводов БИС
ФТЛ	Согласование КДК или АК с трансформатором гальванической развязки	Ретрансляция с восстановлением уровня и крутизны фронтов сигналов	—	1	100	5±0,5	ТТЛШ	24
КДК	Преобразование униполярного последовательного и фазоманипулированного кодов	Распознавание команд-данных, контроль четности, программирование разрядности слов	2—28	1	10 (дин.) 0,1 (стат.)	5±0,5	КМОП	42
АК	Стыковка мультиплексного канала с 8- или 16-разрядной параллельной шиной, преобразование униполярного и фазоманипулированного кодов	Распознавание адреса в канале, контроль слов, распознавание команд-данных, команд общего режима	16	16 8	10 (дин.) 0,1 (стат.)	5±0,5	КМОП	48
СА	Распознавание адреса ВУ в параллельном канале, выработка сигналов чтения-записи устройств ввода-вывода.	Формирование сигнала синхронизации пассивного устройства	—	12 (адрес УВВ)	6 (дин.) 0,1 (стат.)	5±0,5	КМОП	42
МПП	Электрическое согласование уровней сигналов на выходе АК и общей шины, буферизация данных	Контроль по четности слов в общей шине	8	8	10 (дин.) 0,1 (стат.)	5±0,5	КМОП (К588)	28
					150	5±0,5	ТТЛШ (К584)	48
УО	Нормирование сигналов на выходе трансформатора ТИЛ-1В.	Фильтрация низкочастотных помех	—	1	10	5±0,5 —5±0,5	ТТЛ	**14**

лом обмоток — 4, коэффициентом трансформации 1:1:1:1 или 2:1:1:1, намоткой на феррит, числом выводов корпуса, равным 8. Оконечный усилитель — схема, собранная на десяти дискретных компонентах.

БИС УО нормирует уровни электрических сигналов, выполняет роль фильтра нижних частот и согласовывает по входу СТЛ с трансформатором ТИЛ-1В.

БИС КДК (рис. 3; табл. 3). В состав БИС КДК входят независимые декодер и шифратор, имеющие один общий вход начального сброса (MR). Декодер осуществляет прием униполярного фазоманипулированного кода от ФТЛ, модуляцию его частотой 12 МГц, распознавание типа синхроимпульса и контроль двух первых битов данных, после чего выдаются сигналы C или D и TD. На выход DO поступает преобразованный униполярный код, синхронизируемый соответствующей частотой (вывод CO). Все слова контролируются по правильности манчестерского кода, числу битов данных, четности, минимальной длительности пауз и непрерывности.

При передаче шифратор вырабатывает синхроимпульс нужной полярности и вставляет в слово бит четности. Особенность БИС — возможность программирования разрядности информационных слов и полярности бита четности. Это упрощает диагностику мультиплексного канала и повышает его эффективную производительность за счет адаптации к разрядности подключенных источников информации.

Таблица 2

Выводы БИС ФТЛ

Обозначение	Назначение вывода
D00, D01	Входы передатчика
T00, T01, T10, T11	Выходы передатчика
ST	Вход блокировки передатчика
T10, T11	Входы приемника
DI0, DI1, $\overline{DI0}$, $\overline{DI1}$	Выходы приемника
SS	Вход блокировки приемника
RS	Вход схемы контроля питания
R	Выход начального сброса
У	Выход управления при ретрансляции
E	Вход питания (+5 В)
0	Вход земли (0 В)

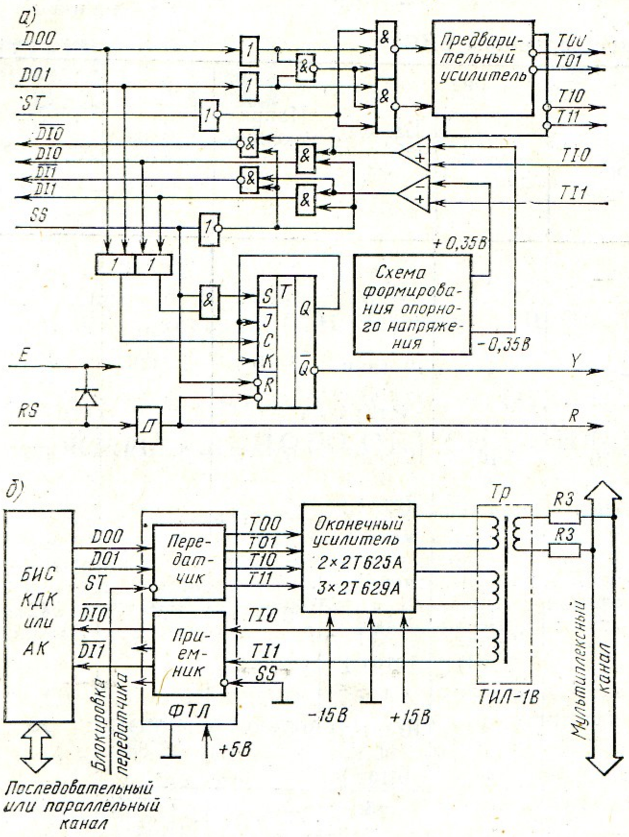


Рис. 2. Структурная схема БИС ФТЛ *(а)* и схема подключения АК или КДК к мультиплексному каналу *(б)*

Предельная тактовая частота синхронизации декодера по входу F — 15 МГц. Это обеспечивает передачу данных в канале со скоростью до 1,25 Мбит/с.

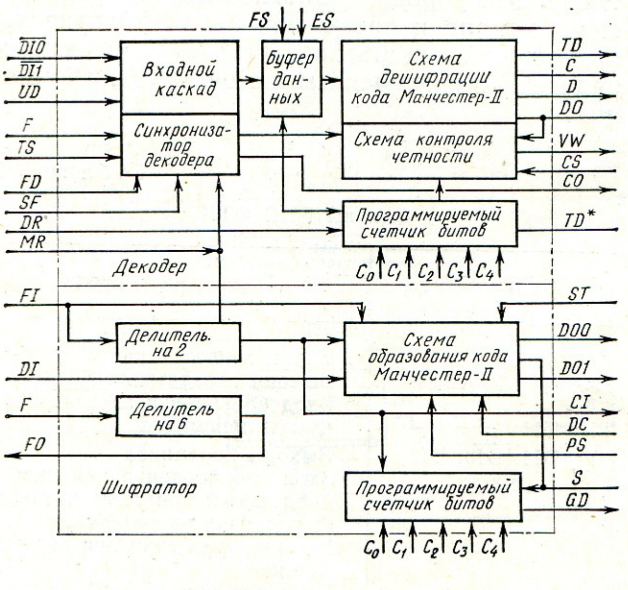


Рис. 3. Структурная схема кодера-декодера манчестерского кода

Таблица 3

Выводы БИС КДК

Обозначение	Назначение вывода
DI0, DI1, UD	Биполярные и униполярный входы данных
F	Входы синхронизации декодера и делителя частоты (12 МГц)
TS, SF	Разрешение синхронизации декодера
DR, MR	Сброс декодера и общий сброс КДК
TD, TD*	Выходы строба выдачи слова (взять слово)
C, D	Стробы приема команд и данных
DO, CO	Выход последовательных данных и их синхронизация
VW	Сигнал правильности слова
CS	Выбор контроля декодера (чет-нечет)
FO	Выход синхронизации шифратора (2 МГц)
FI	Вход синхронизации шифратора (2 МГц)
DI, CI	Вход последовательных данных и частота их синхронизации
DO0, DO1	Выходы фазоманипулированного кода
ST	Запрет выдачи фазоманипулированного кода
DC	Вход задания типа синхроимпульса (команда-данные)
S	Запуск шифратора
PS	Выбор контроля шифратора (чет-нечет)
GD	Выход синхронизации передачи слова (дать слово)
C0, . . ., C4	Входы задания разрядности информации (от 2 до 28 бит)
ES	Разрешение приема синхронизированных данных
FS	Вход синхронизированных данных
FD	Синхронизация синхронного входа FS
E	Вход питания (+5 В)
0	Вход земли (0 В)

БИС АК (рис. 4, *а*; табл. 4). Адаптер канала выполняет сопряжение с 8- или 16-разрядной шиной данных, распознавание адресов абонентов и представляет собой следующий уровень иерархии средств для организации мультиплексных каналов. Адрес абонента мультиплексного канала задается распайкой выводов AD (4—0). При приеме адреса 11111 АК распознает команду общего режима, извещает об этом по выводу OP и подавляет выдачу ответного слова. При задании адреса 11111 БИС переходит в режим монитора и воспринимает данные адресованные всем абонентам. Принимая команды, АК классифицирует их на основной и вспомогательный форматы (подадрес=00000) и извещает систему о типе инструкции по выводу OB. БИС автономно выполняет ряд команд: заблокировать или разблокировать передатчик по выводу ST, установить или сбросить флаг отказа, выдать ответное слово и т. д.

Использование вывода ST позволяет замкнуть в случае необходимости выход декодера и шифратора манчестерского кода, благодаря чему можно подключить АК в «разрыв канала» и наблюдать пересылки в канале на шине данных в режиме монитора.

БИС СА (рис. 4, *б*; табл. 5). Селектор адреса вырабатывает сигналы управления обменом по общей шине для восьми ВУ. На вход BS7 подается низкий потенциал в случае выдачи процессором адреса, относящегося к последним 4К словам адресуемого пространства (DA15, 14, 13-111), закрепленного за ВУ [5]. На входах АВ устанавливается адрес группы УВВ, сравниваемый с кодом на шинах DA12,...A4 общей шины, подключаемых ко входам АА. При их совпадении код АС (DA3, 2, 1) заносится в регистр и дешифруется с выдачей сигналов выборки ВУ на выходах CS7,...,CSO. БИС анализирует сигналы DIN, DOUT, DAO, выдавая строб чтения слова (RD) либо записи старшего и младшего байтов (WR1, WRO). Асинхронный обмен данными организуется по шинам SYNC и RPLY с анализом сигнала готовности YBB (READY).

Магистральные приемопередатчики. В случае необходимости согласования нагрузочной способности выходов АК и общей шины используются магистральные приемопередатчики К588ВА1 ($I_{OL}=8$ мА при $U_{GC}=0{,}5$ В) или К584ВВ1 ($I_{OL}=16$ мА при $U_{OC}=0{,}4$ В или $I_{OL}=53$ мА при $U_{OL}=0{,}8$ В) [6, 7].

Таблица 4

Выводы БИС АК

Обозначение	Назначение вывода
D(15—00)	Параллельная шина данных
OP	Выход индикации приема команд общего режима
OB	Выход индикации приема команд основного (вспомогательного) форматов
SD	Вход запуска передачи данных
SC	Вход запуска передачи команд или ответных слов
KT	Вход задания режима (контроллер-терминал)
AD(4—0)	Входы задания адреса абонента в мультиплексном канале
RD1, RD0	Входы строба считывания старшего и младшего байтов регистров
WR1, WR0	Входы строба записи старшего и младшего байтов регистров
A1, A0	Входы адресной выборки регистров
X1, X0	Входы для подключения БИС к кварцевому резонатору
RT	Вход задания режима ретрансляции данных

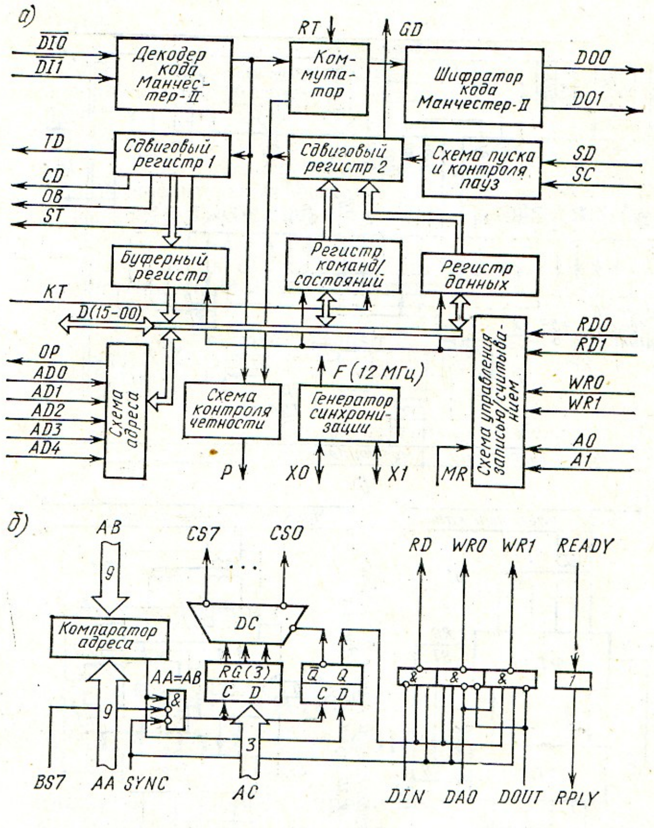


Рис. 4. Структурная схема БИС адаптера каналов (*а*) и селектора адреса (*б*)

Особенности применения БИС комплекта. В простейшем устройстве (рис. 5, *а*), подключенном к общей шине и мультиплексному ка-

Таблица 5

Выводы БИС СА

Обозначение	Назначение вывода
BS7	Вход выбора внешнего устройства
SYNC, DIN, DOUT, AO	Входы управляющих линий общей шины
AA(8—0), AC(2—0)	Входы линий адреса общей шины
AB (8—0)	Входы задания селектируемых разрядов адреса
READY	Вход ответа внешнего устройства
RPLY	Выход «синхронизация внешнего устройства»
RD	Выход «чтение» для внешнего устройства
WR0, WR1	Выходы «запись» младшего и старшего байтов для внешнего устройства
CS7,...,CS0	Выходы «выбор внешнего устройства»
E	Вход питания (+5 В)
0	Вход земли (0 В)

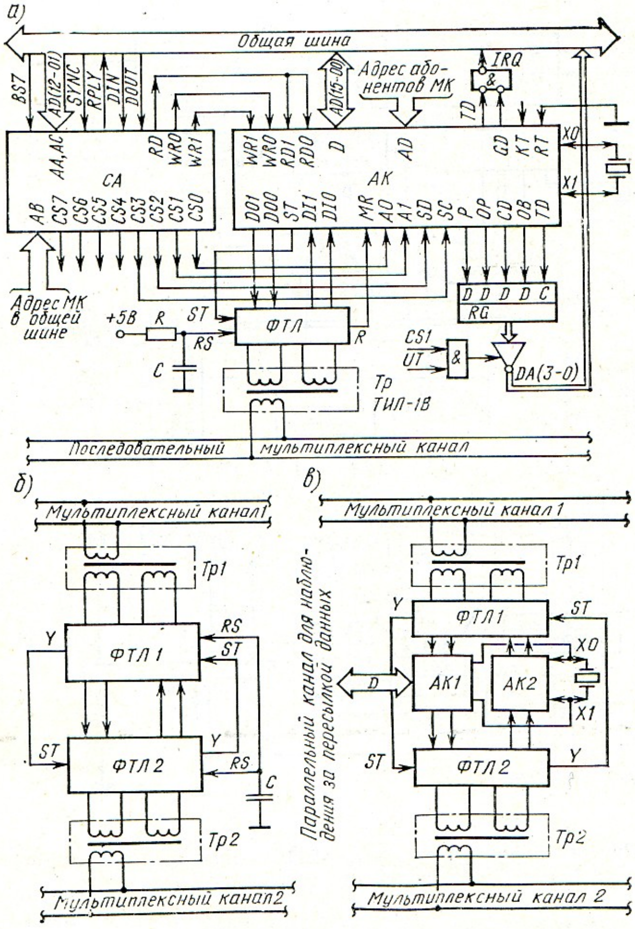


Рис. 5. Примеры применения микропроцессорных БИС: *а* — простейшее интерфейсное устройство; *б* — ретранслятор; *в* — ретранслятор с цифровым восстановлением скважности сигналов

налу, используются СА, АК, ФТЛ и трансформатор ТИЛ-1В. Селектор адреса определяет группу адресов, присвоенных регистрам АК, стробируя их считывание или запись. Обращениями к фиктивным УВВ производится выдача команд, данных или ответных слов в мультиплексный канал по входам SD и SC. При организации взаимодействия микроЭВМ и мультиплексного канала может быть предложено несколько способов извещения системы о необходимости приема или выдачи слов в канал. Один из способов — объединение сигналов TD и GD и подключение их ко входу IRQ БИС контроллера прерываний. Сигналы P, CD, OB и OP могут проверяться процессором при обращении по адресу регистра данных АК на считывание. В случае необходимости, используется прямой доступ в память (при подключении БИС контроллера прямого доступа).

Данная структура может быть адаптирована к каналу микроЭВМ, построенного на основе комплекта БИС К580, благодаря наличию в БИС АК входов WR1, WR0 и RD1, RD0. Другими примерами организации УВВ, подключенных к каналу этого микропроцессора, могут служить схемы, выполненные на базе БИС КДК [8, 9].

Мультиплексные каналы длиной 100 м с 32 шестиметровыми шлейфами — перспективная база для организации комплексных систем. Вместе с тем, в ряде применений требуется организация каналов большей протяженности. Это достигается ретрансляцией сигналов, основа для которой заложена в структуре БИС ФТЛ. Простейший ретранслятор (рис. 5, *б*) может быть построен на двух ФТЛ и трансформаторах. Встроенная в БИС логика устраняет передачу данных в обратном направлении при ретрансляции в результате соединения управляющего вывода (У) со входом блокировки передатчика (ST). Подключение входов схемы контроля питания (RS) через емкость С на землю обеспечивает начальную установку схемы управления в состояние, блокирующее передачу данных в обоих направлениях. Благодаря этому исключается самопроизвольное возбуждение канала.

Ретранслятор — «прозрачная» схема, пропускающая все сигналы в обоих направлениях. Направление, с которого информация приходит раньше, становится главным за счет блокировки встречного направления. Ретранслятор позволяет наращивать длину магистрали или отводящего шлейфа на 100 м, восстанавливать крутизну фронтов и уровень сигналов в канале, что дает возможность строить магистрали произвольной геометрической топологии. При многократной ретрансляции может измениться скважность импульсов. Ее можно восстановить только преобразованием манчестерского кода в униполярный и обратно.

Ретранслятор с цифровым восстановлением скважности сигналов (рис. 5*в*) обеспечивает возможность наблюдения за всеми передачами данных в канале включением в разрыв канала двух БИС АК одновременно. Включение АК вносит дополнительную задержку на 0,5 мкс во время ретрансляции сигналов.

Рассмотренный в статье комплект интерфейсных БИС предоставляет широкие возможности для организации мультиплексных каналов микроЭВМ. В дальнейшем планируется расширить комплект за счет введения в его состав специализированных БИС — контроллеров оконечных устройств, мультиплексного канала и буферизации посылок.

ЛИТЕРАТУРА

1. Sanders L. Pulse codes in serial data communication. — Computer Design, 1982, v. 21, N 1, p. 203—210.
2. Хвощ С. Т. Анализ информационных и временных параметров мультиплексных последовательных каналов. — В кн.: Применение микропроцессорных комплектов БИС при разработке радиоэлектронной аппаратуры. — Л.: ЛДНТП, 1983, с. 17—23.
3. Хвощ С. Т., Шиллер В. А., Кудрявцев В. А., Свиридович В. С. Архитектура комплекта БИС для организации мультиплексных каналов передачи данных. — В кн.: Микропроцессоры и их применение. — Пенза: ПНДТП, 1983, с. 3—4.
4. Хвощ С. Т., Шиллер В. А., Кудрявцев В. А., Горовой В. В. Организация мультиплексных каналов на основе интерфейсного комплекта БИС. — В кн.: Применение микропроцессорных комплектов БИС при разработке радиоэлектронной аппаратуры. — Л.: ЛДНТП, 1983, с. 24—31.
5. Малые ЭВМ и их применение/Под ред. Б. Н. Наумова. — М.: Статистика, 1980. — 231 с.
6. Хвощ С. Т., Горовой В. В., Красницкий Б. М. Опыт использования БИС серии К584 при разработке систем автоматического управления. — Л.: ЛДНТП, 1982. — 26 с.
7. Бобков В. А., Шиллер В. А. Комплект КМДП БИС для специализированных 16-разрядных микровычислительных систем с унифицированным интерфейсом. — Электронная промышленность, 1981, № 4, с. 32—35.
8. Sanders L. For data-communication links, Manchester chip could best. — Electronic Design, 1982, v. 30, N 16, p. 201—210.
9. Williams D. Industrial controller joins the MIL-STD-1553 BUS. — Electronic Design, 1982, v. 30, N 21, p. 205—211.

Статья поступила 18 апреля 1984 г.

УДК 621.3.049.77:681.325.5

В. А. Бобков, М. М. Бобровницкий, В. В. Горовой, В. А. Евдокимов

МИКРОПРОЦЕССОРНЫЙ КОМПЛЕКТ БИС СЕРИИ КА1808 И КОНТРОЛЛЕР ЗЕРКАЛЬНЫХ ФОТОКАМЕР НА ЕГО ОСНОВЕ

Из комплекта БИС КА1808, созданного для автоматизации устройств, приборов и систем бытовой радиоэлектронной аппаратуры и техники, изготовлен контроллер для управления зеркальной фотокамерой, который автоматизирует почти весь процесс съемки — остается только навести на резкость и нажать на кнопку. Автоматизация фотосъемки позволяет за счет одинакового уровня плотности отснятой фотопленки организовать групповую обработку фотоматериалов. При этом экономятся фотореактивы и серебро, значительно сокращается время получения готового снимка.

Как правило, отечественные микропроцессорные комплекты МПК БИС рассчитаны на широкое применение в вычислительной технике, для автоматизации промышленного производства и научных исследований. Поэтому весьма актуально создать МПК для автоматизации различных устройств, приборов и систем бытовой техники. Одним из таких комплектов, в определенной степени восполняющих имеющийся пробел, и является МПК БИС серии КА1808. К МПК БИС для бытовой радиоэлектронной аппаратуры и техники предъявляются обычно требования, обеспечивающие массовость применения: малая разрядность (4 или 8); высокая технологичность и большая серийность.

Микропроцессорный комплекс БИС КА1808

В комплект БИС входят микросхемы для автоматизации процесса сбора, обработки и вывода цифровой информации: КА1808ИР1— аналого-цифровой преобразователь (АЦП), КА1808ВМ1 — центральный процессор (ЦП), КА1808ВУ1 — микропрограммное устройство управления семисегментным индикатором (ДФ), КА1808ВВ1 — схема синхронизации и управления (СУ). Их основные характеристики даны в таблице.

Температурный диапазон работы БИС—25 —+55°С. Общая мощность потребления комплекта — 150 мВт (возможно автономное питание от малогабаритной батарейки). У всех микросхем корпуса пластмассовые с планарным расположением выводов.

Аналого-цифровой преобразователь КА1808ИР1 — 8-разрядный АЦП. Он преобразует аналоговый сигнал (напряжение) по одному из двух входов, имеющих различный наклон передаточной характеристики, в цифровой последовательный код, стабилизирует напряжение и передает спецпризнаки («Готовность фотовспышки», «Тип фотовспышки», «Контроль питания ниже нормы»).

АЦП (рис. 1) содержит аналоговый ключ (АК), интегратор (АИ), компаратор (К), блок контроля (БК) для получения сигнала о состоянии источника питания; источник стабилизированного напряжения (ИСН) для получения опорного и стабилизированного напряжения; операционный усилитель (▷М), распределитель импульсов синхросигнала (РИ), схему управления (СУ), 8-разрядный счетчик (СЧ), выходной мультиплексор (М).

Таблица

Основные характеристики МПК БИС серии КА1808

Параметр	КА1808ИР1	КА1808ВМ1	КА1808ВУ1	КА1808ВВ1
Разрядность	8	8	8	—
Частота микросигнала, кГц	32	32	32	32
Способ организации управления	Синхронный	Синхронный	Синхронный	Синхронный
Технология	ТТЛ, И²Л	И²Л	ТТЛ, И²Л	ТТЛ
Тип корпуса	4109.20-1	4117.22-3	4117.22-3	4109.20-1
Число выводов	20	22	22	20
Напряжение питания, В	$5{,}0^{+10}_{-20}\%$	$5{,}0^{+20}_{-5}\%$	$5{,}0^{+10}_{-20}\%$	$5{,}0^{+10}_{-20}\%$
Максимальный ток потребления, мА	10	22	8	10
Ток нагрузки по выходам, мА	0,1	0,1	0,8	1,0 — по выходам 6,13 10 — по выходам 1, 2, 17, 18

Примечание: напряжение 1,5 В вырабатывает МС КА1808ИР1.

АЦП синхронизируется распределителем импульсов. Входной сигнал преобразуется в цифровой код методом двухшагового интегрирования [2].

По сигналам из схемы управления аналоговый ключ подключает к интегратору один из двух входных сигналов или источник опорного напряжения. Компаратор выдает сигнал сравнения проинтегрированного и опорного напряжения.

8-разрядный счетчик формирует цифровой эквивалент входного сигнала. Выходной мультиплексор выдает в последовательном коде признаки и цифровой эквивалент входного аналогового сигнала. В АЦП постоянно контролируется питающее напряжение. При понижении напряжения питания ниже нормы на выходе АЦП появляется соответствующий сигнал.

Центральный процессор КА1808ВМ1 — 8-разрядный последовательный. Он реализует алгоритм работы фотокамеры; вычисляет по принимаемым значениям освещенность и чувствительность пленки, величины выдержки и (или) диафрагмы и выдает сигналы управления на исполнение. БИС ЦП (рис. 2) содержит программируемую логическую матрицу (ПЛМ), блок модификации адреса (БМА), регистр адреса микрокоманды (РАМК), регистр хранения кода освещенности (РВ), мультиплексор входных данных (М), блок формирования сигналов управления (БФСУ), преобразователь кодов (Х-У), арифметическо-логическое устройство (АЛУ), блок индикации (БИ), блок отработки (БО), счетчик (СТ), схема синхронизации (СС). Взаимодей-

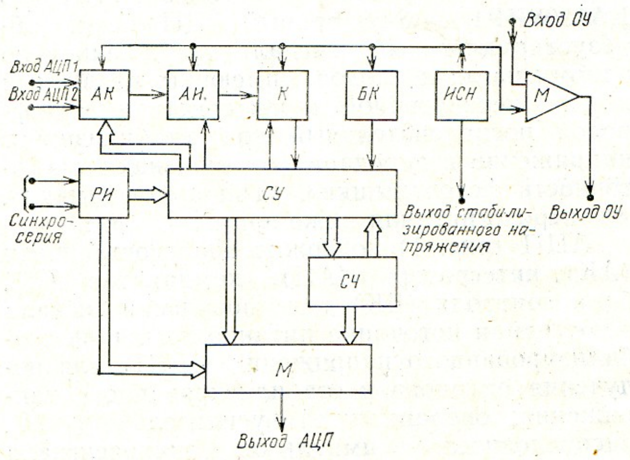


Рис. 1. Структура БИС КА1808ИР1

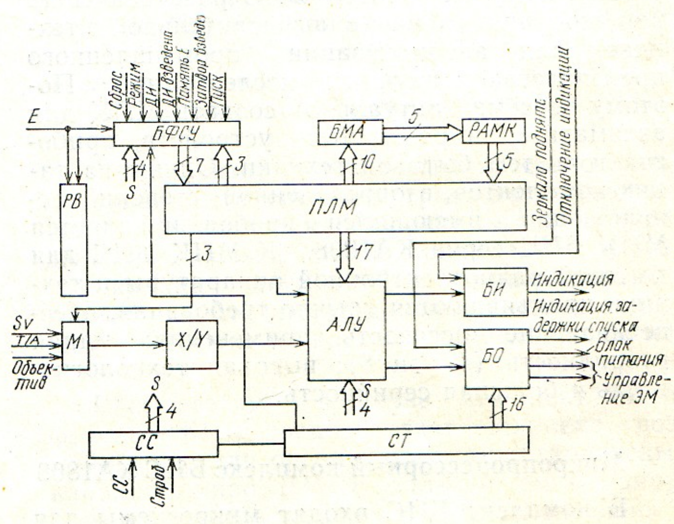


Рис. 2. Структура БИС КА1808ВМ1

ствием всех блоков управляет микропрограммный автомат, реализованный на ПЛМ, БМА, РАМК.

Входные данные о чувствительности пленки, выдержке, диафрагме и объективе представляются в коде Грея (для надежности ввода информации с механических наборников кодов). Экспонометрические параметры вычисляются по основной экспонометрической формуле [1].

Рассчитанные значения выдержки и диафрагмы либо передаются на индикацию БИ, либо отрабатываются посредством БО.

Микропрограммное устройство управления семисегментным индикатором (ДФ) КА1808ВУ1 (рис. 3) содержит схемы ввода данных (СВД) и синхроимпульсов (СВС), схему сброса (СС), 14-разрядный входной регистр (ВР), схему управления (СУ), 7 катодных ключей (КК), мультиплексор (М); программируемую логическую матрицу (ПЛМ), 7-разрядный выходной регистр (РВ), схему формирования спецсимволов (СФСС), 7 анодных ключей.

Основные функции БИС ДФ — прием с центрального процессора КА1808ВМ1 кодов выдержки, диафрагмы и спецсимволов; организация высвечивания выдержки, диафрагмы и спецсимволов на светодиодном индикаторе; опрос датчиков кодов и подачи в центральный процессор кода установленного режима работы фотоаппарата.

Особенность БИС КА1808ВУ1 — организация динамической индикации светодиодного индикатора. Ток нагрузки выходных каскадов — 0,8 мА.

БИС синхронизации и управления (СУ) КА1808ВВ1 используется для выработки синхросигнала управления исполнительными механизмами и индикации состояния источника питания.

БИС СУ (рис. 4) содержит: генераторы контроля питания (Г1) и прямоугольных импульсов синхронизации (Г2), коммутатор (К), схему формирования сигнала начала работы фотокамеры (СС), схемы блокировки питания (СБП), коррекции тока нагрузки (СКТН), формирование сигналов управления светодиодом (СУ СД) и электромагнитами (СУ ЭМ) и сигнала начала отработки выдержки (СНВ).

Электрические характеристики БИС СУ регулируются при подключении внешних резисторов. Частота генератора Г2 изменяется от 20 до 100 кГц в зависимости от номинала резистора между 9 и 14 выводами. Ток нагрузки по выходам 2, 17, 18 изменяется от 5 до 25 мА в зависимости от номинала резистора между 10 и 11 выводами микросхемы.

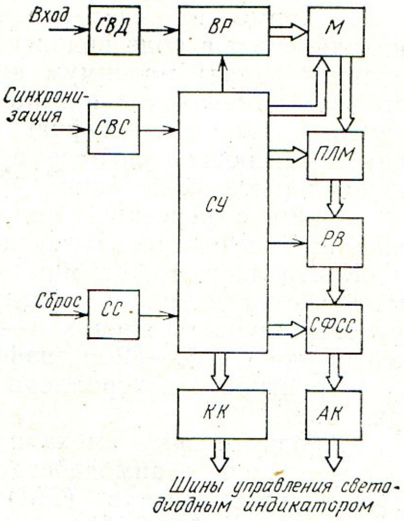


Рис. 3. Структура БИС КА1808КУ1

БИС СУ состоит из отдельных функционально-законченных блоков. Каждый блок БИС можно использовать самостоятельно в радиотехнических устройствах различных классов.

Контроллер на базе МПК серии КА1808

Контроллер обеспечивает следующие режимы работы зеркальных фотокамер: автомат выдержки; автомат диафрагмы; программный режим выбора экспозиции (пара выдержка-диафрагма выбираются по линейному закону до полного открытия диафрагмы, с продолжением регулирования по выдержке); автоматический выбор экспозиций с репетиром и с лампой-вспышкой; запись и хранение в памяти условий экспозиции.

Вспомогательные режимы: автоматическая задержка спуска на 2 и 10 с с индикацией приближения момента спуска; работа с моторным приводом перемотки; контроль питания.

Режимы работы и вычисленные экспонометрические величины индицируются в циф-

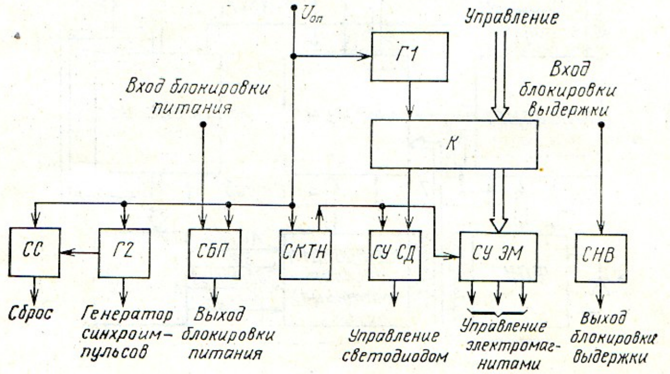


Рис. 4. Структура БИС КА1808ВВ1

ровой форме на светодиодном индикаторе, находящемся в поле видоискателя. Цифровой индикатор дает максимум информации о величинах диафрагмы и выдежки, о сигналах «ручная установка диафрагмы», «фотовспышка» и мигающем сигнале предупреждения о неправильной экспозиции.

Режим с репетиром позволяет использовать объектив без «прыгающей» диафрагмы.

Контроллер может обрабатывать экспонометрические величины в следующих пределах: чувствительность пленки 8—16000 ед. ГОСТ; выдержка 1/1000—32 с; диафрагма 1,2—22.

Структура контроллера приведена на рис. 5.

Исполнительные механизмы фотокамеры и контроллер взаимодействуют посредством трех электромагнитов (ЭМ) и восьми микропереключателей. ЭМ1 управляет подъемом зеркала; ЭМ2 — диафрагмой объектива и ЭМ3 — второй шторкой затвора.

Микропереключатели подключаются непосредственно к соответствующим выводам БИС. Положение микропереключателей (вкл.-выкл.) определяет состояние исполнительных механизмов (затвор взведен, зеркало поднято) и задает режимы фотокамеры: просмотр и запоминание условий экспозиции; отключение индикации; подключение синхроконтакта фотовспышки; контроль питания, спуск затвора.

Для настройки фотокамеры достаточно произвести три регулировки: резистором R1 выставить наклон передаточной характеристики тракта ФПУ-АЦП, равный 18 мВ/ступень; резистором R2 установить верхний порог чувствительности фотоприемника, а резистором R3 — время подъема зеркала.

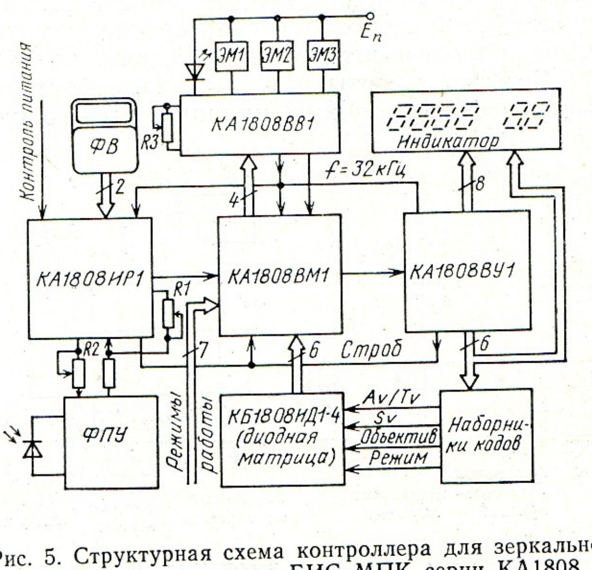


Рис. 5. Структурная схема контроллера для зеркальной фотокамеры на основе БИС МПК серии КА1808

Суммарная погрешность измерения и расчетов, выполняемых контроллером, не превышает $^1/_8$ ступени.

Благодаря функциональной законченности БИС серии КА1808, простоте управления, электрическим характеристикам, они могут успешно применяться в широком классе радиотехнических устройств и в устройствах автоматики и телемеханики.

ЛИТЕРАТУРА

1. Джеймс Т. Х. Теория фотографического процесса/Пер. с англ. 2-е рус. изд. — Л.: Химия, 1980.

2. Титце У., Шенк К. Полупроводниковая схемотехника/Пер. с нем. под ред. А. Г. Алексенко. — М.: Мир, 1982.

Статья поступила 31 мая 1984 г.

РЖ ВИНИТИ

Перспективы замены больших ЭВМ микромашинами. Junking the mainframe. Healey Martin. **«Datamation», 1983, 29, № 8 (англ.)**

Отмечается, что появление в 70-е годы мини-ЭВМ не повлияло на доминирующую роль больших ЭВМ в ВТ. Прогнозируется, что вытеснение больших ЭВМ произойдет постепенно; через 5 лет микроЭВМ будут преобладать, а через 10 лет большие ЭВМ полностью исчезнут как категория ВТ. Выделяются и рассматриваются основные тенденции, способствующие этому процессу.

Ключевым моментом эволюции ВТ в сторону микро-ЭВМ и ВС является прогресс в области повышения уровня интеграции схем. СБИС существенно сокращают сроки и стоимость новых разработок. МП повышают уровень программируемости периферийных устройств; микро-ВС будут объединяться в локальные сети, в свою очередь объединяемые в неоднородную иерархическую сеть. Ил. 3. [Univ. of Cardiff, Великобритания].

Применение ЭВМ в исследованиях и разработках. Computers in research and development. Wilson B. L. H. «Radio and Electron. Eng.», 1983, 53, № 10 (англ.)

В обзоре, ориентированном на разработчиков полупроводниковых приборов, не являющихся специалистами в области ВТ, рассматриваются методы применения ЭВМ, главным образом МП, для исследований и разработок. В частности, обсуждается использование ЭВМ в системе машинного проектирования и изготовления матриц К/МОП-вентилей семейства Plessey CLA 2000.

В этом случае ЭВМ осуществляет преобразование спецификаций разработчика в специальную последовательность команд, по которой изготавливается фотошаблон будущей ИС. ЭВМ помогает проанализировать электрические параметры ИС для различных вариантов топологий, построить графики распределения напряжений, концентрации носителей с помощью различных моделирующих программ, а также провести испытание ИС. [Plessey Res. (Caswell) Ltd., Великобритания].

УДК 681.3.022

А. А. Козак, С. И. Сорока

УСТРОЙСТВА ОТОБРАЖЕНИЯ ИНФОРМАЦИИ МИКРОПРОЦЕССОРНЫХ СРЕДСТВ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Использование микропроцессоров и микроЭВМ в устройствах отображения информации позволяет создавать модели различного уровня сложности от простейших видеомониторов и индикаторных дисплеев на основе матричных индикаторных панелей, обеспечивает увеличение надежности, расширение функциональных возможностей, уменьшение габаритных размеров, потребляемой мощности и повышение технологичности этих устройств в условиях массового производства.

Устройства отображения информации (УОИ), обеспечивающие оперативное информационное взаимодействие оператора с микроЭВМ являются одним из основных элементов систем на базе МСВТ. Современные эффективные УОИ по своему назначению должны соответствовать УОИ традиционной вычислительной техники, но отличаться от них развитыми функциональными возможностями, высокой надежностью, малыми габаритными размерами и потребляемой мощностью, низкой стоимостью и повышенной технологичностью в условиях массового производства. Для таких УОИ характерно многообразие технических средств и физических принципов, используемых в реализации функции отображения информации; расширение области применения, благодаря универсальности, адаптивности и дешевизне МСВТ; стремление совместить в одном изделии возможности отображения, обработки и ввода данных.

Из-за отсутствия единого подхода к определению и классификации общепринятая терминология в области УОИ не установилась, границы класса УОИ определены нечетко, а существующие классификации этих изделий, предлагаемые различными авторами [1—3], часто не соответствуют и даже противоречат друг другу. В связи с этим уточним некоторые понятия и определения.

УОИ МСВТ — периферийные устройства микроЭВМ, содержащие экран, средства записи, хранения и представления данных в удобной форме и осуществляющие оперативную информационную связь пользователя с задачей, решаемой при помощи программно-технических комплексов МСВТ.

Экран — оптически реверсивная среда, обеспечивающая визуализацию сигналов, поступающих от пользователя и устройств, участвующих в решении задачи.

Классификация УОИ МСВТ (рис. 1) составлена с учетом системного распределения функций в комплексе «УОИ-микроЭВМ». В соответствии с характером пользования и числом пользователей УОИ МСВТ можно разделить на устройства индивидуального, коллективного и массового пользования. На данном этапе развития МСВТ больше внимания уделяется созданию УОИ первой группы (дисплеев), непосредственно подключаемых к микроЭВМ. Внутри каждой группы в соответствии с характером связи пользователей с задачей, решаемой на микроЭВМ, можно выделить устройства вывода информации экранные (УВИЭ) и устройства ввода-вывода информации экранные (УВВИЭ).

Видеомониторы — УВИЭ, осуществляющие преобразование электрических сигналов (как правило, телевизионных) в изображение на экране.

Индикаторные дисплеи — УВИЭ, осуществляющие преобразование последовательности цифровых кодов в изображение на экране и содержащие, как правило, запоминающее устройство и генератор элементов изображения.

Диалоговые дисплеи — УВВИЭ, содержащие индикаторный дисплей, и обеспечивающие возможность ручного ввода и модификации пользователем отображаемых данных.

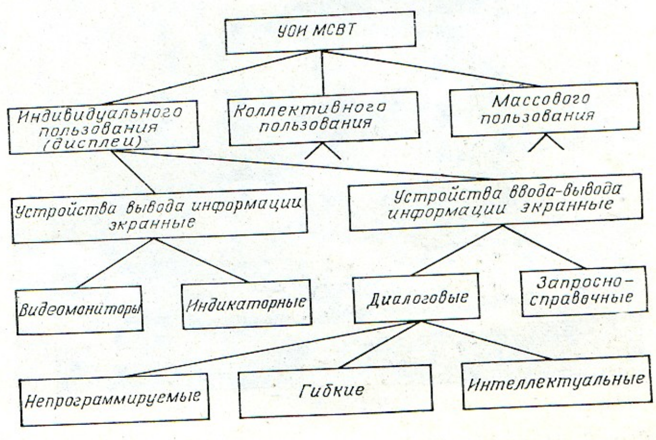


Рис. 1. Классификация УОИ МСВТ

Запросно-справочные дисплеи отличаются от диалоговых отсутствием возможности модификации отображаемых данных.

По возможности программирования различают **дисплеи с программируемыми функциями и параметрами** (гибкие); **дисплеи с программируемой обработкой данных** (интеллектуальные); **непрограммируемые дисплеи.**

В зависимости от вида представляемой информации дисплеи делятся на **алфавитно-цифровые** (символьные), **знакографические** (т. е. алфавитно-цифровые, обладающие графическими возможностями), **графические** и **полутоновые.** В последних информация представляется при помощи градаций яркости каждого элемента разложения рабочего поля экрана.

По способу формирования изображения дисплеи делят на **растровые**, информация в которых представляется изменением интенсивности светового потока при последовательной адресации всех элементов рабочего поля экрана или заранее определенной его части, и **координатные** (функциональные) — с информацией, формируемой, как правило, из отдельных линейных отрезков (векторов) **посредством** адресации только в участки рабочего поля экрана, соответствующие отображаемым элементам.

Термины одноцветный или многоцветный, запоминающий или регенерационный дисплей — в пояснениях не нуждаются.

К основным характеристикам УОИ МСВТ относятся следующие:

— размер рабочего поля индикатора (экрана);

— информационная емкость экрана (число строк и знаков в строке, число элементов разложения рабочего поля экрана по каждой координате);

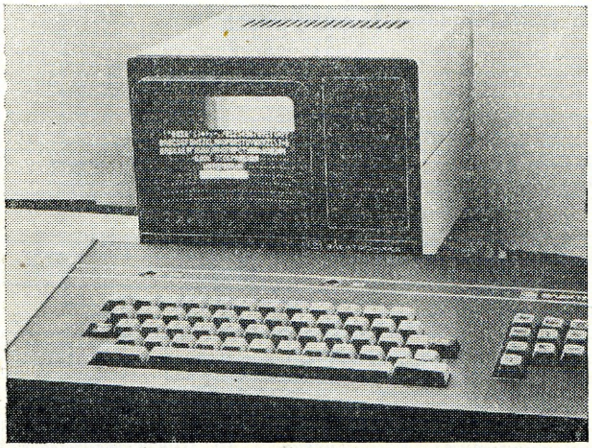


Рис. 2. Индикаторный дисплей на ЭЛТ с диагональю 16 см УВВК-16-001

— форматы отображаемых знаков и матрицы знаков;

— число и типы знаков и графических элементов в наборе;

— скорость построения векторов или знаков;

— емкость буферного ЗУ;

— эргономические параметры изображения (размеры знаков и интервалов, яркость, контрастность и т. д.);

— наличие и характер расширенных функциональных возможностей;

— массогабаритные и энергетические показатели;

— показатели надежности;

— число отображаемых цветов, цветовых оттенков, полутонов;

— тип и характеристики интерфейса;

— тип и характеристики клавиатуры.

Современное состояние УОИ МСВТ. По назначению и областям применения УОИ МСВТ можно разделить на предназначенные для комплектации универсальных микроЭВМ и для комплектации технологического оборудования, АСУ, САПР и т. д.

В соответствии с современной концепцией МСВТ разработчики дисплеев стремятся уменьшить их габаритные размеры и массу. Однако с учетом эргономических требований и характером зрительного восприятия информации размеры экрана нельзя непрерывно уменьшать без потери объема отображаемой информации. Поэтому снижение массогабаритных параметров дисплеев достигается не столько благодаря применению малоформатных экранов, сколько широкому использованию микропроцессорной техники, которая принимает на себя функции формирования и редактирования отображаемых на экране картин при помощи имеющихся в составе некоторых микроЭВМ дисплейных адаптеров, управляющих видеомониторами. Таким образом, микропроцессорные средства по отношению к УОИ МСВТ играют, с одной стороны, роль источника информации, а с другой — универсального блока управления отображением. Дисплеи МСВТ можно определить как аппаратно-программные средства, в которых *аппаратно* реализуется управление отображением, формирование видеосигнала, регенерация информации, а *программно* — обработка команд пользователя.

В настоящее время серийно выпускаются видеомониторы растрового и координатного типов, индикаторные дисплеи, диалоговые алфавитно-цифровые и знакографические дисплеи. Наиболее широко используются простые видеомониторы, которые выполняются как в виде автономных изделий, так и в виде составной части индикаторных, запросно-спра-

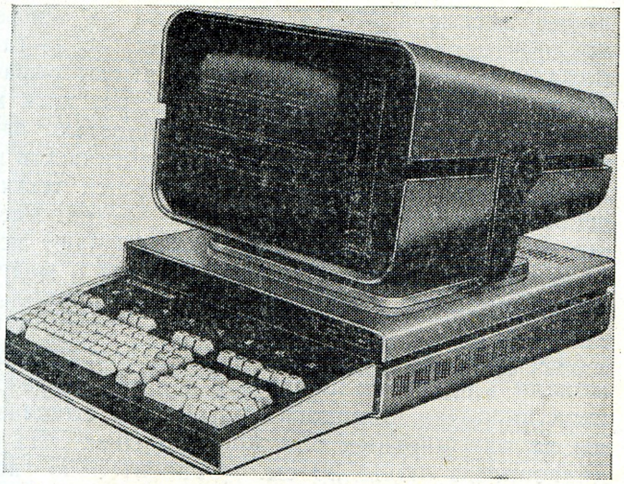


Рис. 3. Диалоговый алфавитно-цифровой дисплей 15ИЭ-00-13

вочных, диалоговых дисплеев и диалоговых вычислительных комплексов (ДВК). Для отображения алфавитно-цифровой, графической и полутоновой информации, а также управления технологическими процессами выпускаются видеомониторы (ПОЭ.ИФА-16) и индикаторные дисплеи на ЭЛТ с диагональю экрана 16 см (УВВК-16-001) и 31 см (15ИЭ200-140-018). В дисплее УВВК-16-001 (рис. 2) обеспечивается одновременное отображение натурных видеоданных (от телекамеры) об управляемых процессах и текстовой информации (от микроЭВМ).

Для комплектации универсальных микроЭВМ и систем на их основе предназначены диалоговые алфавитно-цифровые дисплеи 15ИЭ-00-13 (рис. 3) и 15ИЭ200-140-017 (рис. 4). Число строк и знаков в строке для этих дисплеев равно 24×80 при формате матрицы знака 7×8 элементов для первого и 7×9 — для второго. Набор отображаемых символов — русские и латинские буквы (строчные и прописные), цифры, специальные символы. В настоящее время осваивается более гибкий алфавитно-цифровой дисплей широкого применения МС 6102, адаптируемый к различным микроЭВМ и легко модернизируемый (рис. 5). Размер рабочего поля экрана дисплея равен 31 см, число строк и знаков в строке — 24×80, формат матрицы знака — 7×9, потребляемая мощность — 140 Вт, интерфейсы — ИРПС, ИРПР, С2.

К основным преимуществам дисплея МС6102 относятся: возможность программирования функций пользователем; способность к отображению 16 различных полей на экране; наличие элементарной графики; модульная структура, позволяющая наращивать функциональные возможности (до создания ДВК с подключением периферии через стандартные порты-выводы); расширенные функции редактирования (вставка знака, строки, изъятие знака, строки); большой набор символов; двухстраничная память.

Основным типом индикаторов для дисплеев МСВТ продолжает оставаться ЭЛТ, обладающие достаточным быстродействием, высокой светоотдачей, имеющие небольшое число выводов и управляющих схем, отработанность технологии изготовления, сравнительно низкую стоимость. Однако такие недостатки ЭЛТ, как необходимость преобразования дискретных сигналов в аналоговые, недостаточная стабильность изображения, высокие значения питающих напряжений, объемность конструкции, невысокая надежность делают дисплеи на ЭЛТ плохо приспособленными для миниатюризации и трудносовместимыми с МСВТ.

В связи с этим в последние годы наблюдается повышенный интерес к знакосинтезирующим индикаторам (ЗСИ), изображение в которых формируется (синтезируется) из совокупности дискретных элементов, расположенных в одной плоскости. ЗСИ отличаются высокой надежностью, точностью и линейностью отображения, малой глубиной конструкции, большим углом обзора, простотой совмещения оптического (статического) и электронного (динамического) изображений. Каждый дискретный элемент ЗСИ имеет жесткий адрес, что позволяет эффективно использовать плоские матричные индикаторы в сочетании с микроЭВМ [4].

За последние десять лет заметные успехи достигнуты в области создания газоразрядных индикаторных панелей (ГИП). Дисплеи

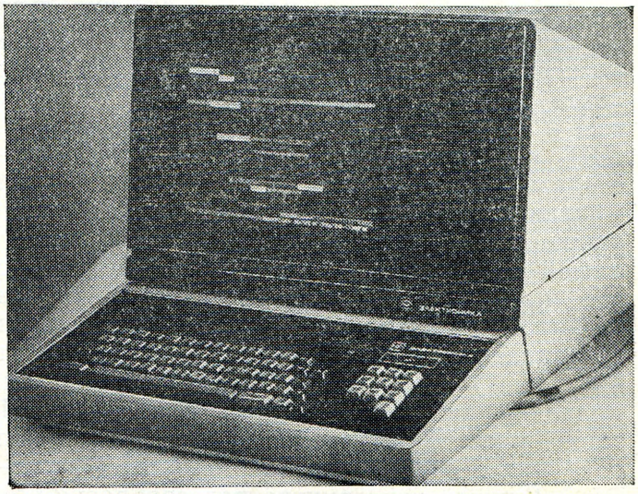


Рис. 4. Диалоговый алфавитно-цифровой дисплей 15ИЭ 200-140-017

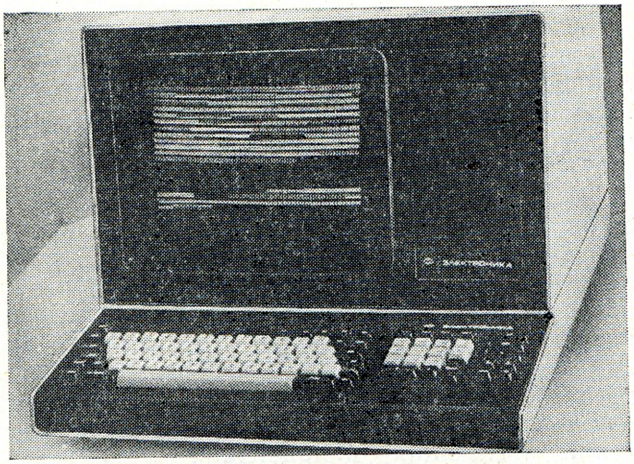


Рис. 5. Программируемый алфавитно-цифровой дисплей МС6102

на ГИП (рис. 6) уменьшились в объеме почти в 15 раз, потребление энергии снизилось в 10 раз. Число адресуемых элементов дисплея составляет 512×512, разрешающая способность — 20 лин/см, цвет свечения — оранжево-красный, яркость свечения — 100 кд/м².

Основным препятствием для широкого использования ГИП в дисплеях МСВТ является их высокая стоимость (в несколько раз выше стоимости ЭЛТ такой же информационной емкости экрана), неравномерность распределения свойств панели по рабочему полю экрана, сложность и большое число управляющих схем. Кроме этого, для некоторых применений неприемлема дискретность отображения, присущая ЗСИ.

В настоящее время наметились два направления решения проблемы комплектации универсальных микроЭВМ и создания ДВК на их основе. Первое — разместить в одном конструктиве наибольшее число частей комплекса вместе с микропечатающим устройством и НМЛ. Такой подход использован в ДВК «Электроника НЦ-80-20/3». Второе — создать семейство дисплеев на основе базовой модели и принципа модульности. Последняя концепция применена при разработке гибкого алфавитно-цифрового дисплея МС6102 (базовая модель), в котором предусмотрена возможность расширения его функций путем встраивания дополнительных модулей (печатных плат) и оснащения интерфейсами для подключения периферийных устройств. Из зарубежных фирм сторонником первого направления является фирма HP, второго — DEC. Единый конструктив целесообразно применять в том случае, когда ограничены объемы и стоимость изделия. Использование принципа модульности и базовой модели при создании дисплеев МСВТ и ДВК на их основе предпочтительнее во всех случаях, когда габариты и стоимость изделия не имеют решающего значения. Такая архитектура, хотя и связана с определенной избыточностью, однако обеспечивает лучшую технологичность и унификацию изделий за счет бо́льшей технической однородности отдельных частей. Рассмотренные направления не исключают использования унифицированных узлов и модулей (например, индикаторных, вычислительных и т. п.).

Современные отечественные и зарубежные модели алфавитно-цифровых и знакографических дисплеев широкого применения выполнены на одноцветных или многоцветных ЭЛТ с диагональю около 30 см. Относительно небольшие размеры экрана дают возможность реализовать простые, ремонтопригодные конструкции дисплеев, обеспечивающие повороты и наклоны индикатора по желанию пользователя.

Безусловно одним из важнейших свойств современных дисплеев МСВТ является возможность программирования их характеристик пользователем. Гибкое программирование атрибутов текста и изображения может осуществляться с помощью микропроцессоров как самим пользователем, так и программным путем.

Удовлетворение потребности в УОИ для интерактивных графических систем традиционно решается при помощи дисплеев, использующих координатный (функциональный) способ формирования изображения. В этих устройствах применяются как обычные, так и запоминающие ЭЛТ. В первом случае лучше обеспечивается интерактивный режим, во втором — вывод на экран больших объемов отображаемой информации. Важные достоинства дисплеев на запоминающей ЭЛТ — полное отсутствие мельканий и идеальная пространственная стабильность выводимого изображения. Существенные недостатки таких дисплеев обусловлены сложностью селективного стирания, указания световым пером, получения многоцветного изображения.

Для использования в САПР предназначен видеомонитор 15ИЭ-162-210-001 на базе запоминающей ЭЛТ с диагональю 31 см. Скорость построения изображения равна 100 м/с, время хранения отображаемой информации не менее 15 мин., толщина линии не более 0,8 мм.

Всеми достоинствами координатных (функциональных) устройств отображения (высокая разрешающая способность, непрерывность отображаемых линий и др.) обладают дисплеи с ЭЛТ типа пенатрон, однако они сложны в изготовлении и эксплуатации, а для управления цветом требуется высоковольтный быстродействующий переключатель.

Недостатки координатных дисплеев, в том числе невозможность отображения многоцветной информации, вызвали необходимость разработки растровых крупноформатных дисплеев. Этому способствуют успехи в создании высококачественных масочных кинескопов, БИС-контроллеров, эффективных алгоритмов преобразования сигналов цифрового кодирования в сигналы управления подсветом ЭЛТ. В настоящее время создаются небольшие (по сравнению с координатными) и простые в управлении растровые графические дисплеи на базе многоцветных ЭЛТ с диагональю 51 см и разрешающей способностью до 1280×1024 элемента (триад).

В системах отображения информации коллективного пользования (СОИ КП) широко применяются ГИП постоянного тока, содержащие 32×32 элемента индикации. СОИ КП в свой состав, как правило, включают экран матричного типа, работающий параллельно с дисплеем МСВТ и нередко имеющий общий с ним контроллер, реализованный на базе микропроцессорных средств. Современные дисплеи МСВТ оснащены клавиатурой, включающей алфавитно-цифровые, калькуляторные, функциональные группы клавиш.

Перспективы развития УОИ МСВТ. Прогресс в области УОИ МСВТ в значительной мере определяется успехами в разработке новых типов индикаторов. Исследования последних лет показывают, что перспективными для применения в дисплеях являются жидкокристаллические, электролюминесцентные и полупроводниковые ЗСИ, а также новый тип индикатора — ЭЛТ с матричным управлением и плазменным катодом [5]. Этот индикатор имеет люминофорный экран, источник электронов и управляющие электроды, заключенные в корпус толщиной до 6 см. Устройство сочетает в себе преимущества ГИП и индикаторов на ЭЛТ. Газовый разряд возникает между задней стороной катода и анодными строчными электродами, расположенными на управляющей пластине. Электроны вытягиваются из плазмы при подаче на избранные столбцы управляющего напряжения и ускоряются при движении к экрану напряжением 4 кВ. Панель, воспроизводящая 240 символов, по быстродействию превосходит ЭЛТ, так как можно одновременно возбуждать группу элементов или всю строку. Отсутствие дрожания и высокая частота кадров обеспечивают высокое качество изображения. Очень короткие пучки электронов исключают расфокусировку по всей площади экрана, включая углы. Для получения цветных изображений применима аналогичная техника ЭЛТ (пятна трех люминофоров).

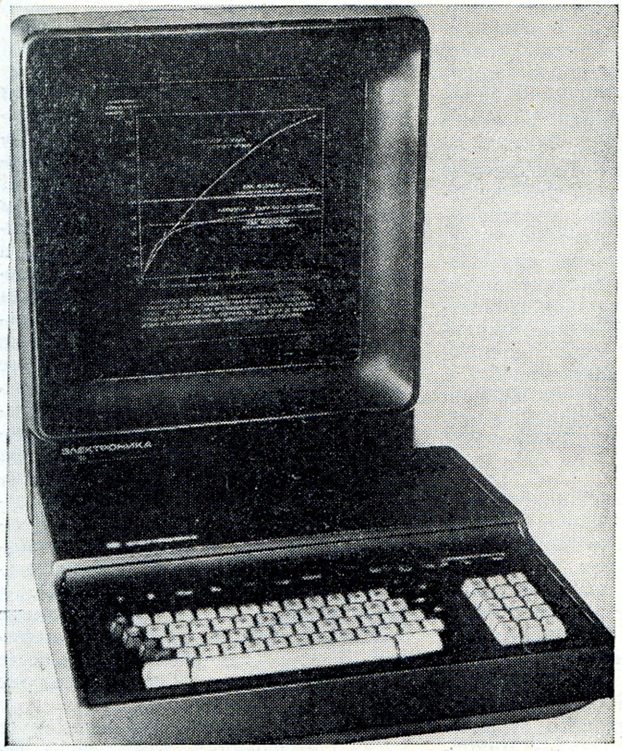


Рис. 6. Диалоговый знакографический дисплей на газоразрядной индикаторной панели переменного тока

Помимо поиска новых типов индикаторов характерной тенденцией является стремление улучшить параметры устройств отображения, и в первую очередь эргономические. Основным стимулом создания дисплеев, удобных для человека, является повышение производительности труда пользователей за счет снятия напряжения и лучшего использования рабочего времени. Главное внимание при этом уделяется обеспечению следующих характеристик:

— поэлементное управление экраном;
— позитивное и негативное изображение;
— отсутствие отражений;
— оптимальный размер, возможность наклона и поворота экрана;
— высокая частота регенерации изображения;
— встроенные диагностические средства;
— выбор приемлемого цвета свечения, многоцветность;
— программируемость (в том числе клавиатуры);
— формат знакоместа и размер символа.

В дисплеях с поэлементным управлением можно осуществлять отображение и обработку графических изображений, причем каждой позиции экрана можно приписать несколько атрибутов (цвет, несколько уровней яркости, мерцание), что создает максималь-

ные удобства пользователю. Однако при реализации поэлементного управления экраном требуется большой объем памяти, средства быстрого обмена данными между памятью и экраном и быстродействующий процессор, изменяющий содержимое памяти. Стоимость указанных средств быстро снижается, что создает предпосылки для оснащения в недалеком будущем всех терминалов вычислительных машин многоцветными дисплеями с высокой разрешающей способностью и расширенными функциональными возможностями. Разумеется, необходимой частью подобных дисплеев является высококачественный многоцветный индикатор.

Процесс создания дисплейной техники затрагивает сложные вопросы разработки архитектуры, конструкции, более современной элементной базы, технических и программных средств для устройств отображения информаций. Это требует комплексного подхода к решению поставленных задач, унификации технических и программных средств, что обеспечит применение дисплеев МСВТ в самых различных областях народного хозяйства.

ЛИТЕРАТУРА

1. Диалоговые устройства отображения информации на электронно-лучевых трубках./Под ред. М. К. Сулима. — М.: Статистика, 1977. — 184 с.
2. Смоляров А. М. Системы отображения информации и инженерная психология. — М.: Высшая школа, 1982. — 272 с.
3. Шерр С. Электронные дисплеи. — М.: Мир, 1982. — 624 с.
4. Кузнецов П. Д., Козак А. А., Безродный М. С. Отображающие устройства ввода—вывода информации микропроцессорных средств вычислительной техники. — Радиотехника, № 1, 1983, с. 36—39.
5. Шауэр А. Плоская 35,5 см индикаторная люминофорная панель с плазменным источником электронов. — Электроника, 1982, т. 55, № 25, с. 25—29.

Статья поступила 27 марта 1984 г.

УДК 681.324

А. А. Мячев

ПРИНЦИПЫ СОГЛАСОВАНИЯ СИСТЕМНЫХ ИНТЕРФЕЙСОВ мини- и микроЭВМ

Анализ системных интерфейсов наиболее массовых мини- и микроЭВМ и устройств их согласования позволил определить технически и экономически оправданные основные области их применения.

В последнее время особое значение придается работам по проектированию систем различной мощности и назначения на базе мини- и микроЭВМ, обеспечивающих единое программное управление различными периферийными устройствами на основе процессорно-независимых интерфейсов [1]. Именно построение мультимикропроцессорных систем (МПС) различного назначения представляется наиболее эффективной сферой применения массовых средств вычислительной техники (СВТ) [2].

Основные проблемы, сдерживающие развитие массовых МПС на данном этапе, связаны с разработкой контроллеров ВЗУ, ПУ, УСО, устройств межсистемной связи и спецпроцессоров, выходящих на системные интерфейсы [2]. При проектировании МПС целесообразно рассмотреть концепцию совместного использования технических средств наиболее распространенных семейств мини- и микроЭВМ на основе устройств согласования интерфейсов (УСИ) совместимых и несовместимых ЭВМ [3].

По функциональному назначению УСИ можно разделить на два основных класса: *гомогенные,* согласовывающие магистрали системно-совместимых ЭВМ; *гетерогенные,* связывающие интерфейсы несовместимых ЭВМ. В каждом классе можно выделить УСИ-мини и УСИ-микро, эмулирующие интерфейсы более мощных моделей ЭВМ (для подключения к микроЭВМ устройств мини-ЭВМ) и микроЭВМ (для подключения к мини-ЭВМ устройств микроЭВМ) соответственно.

В настоящее время наибольшее распространение получили *гомогенные УСИ-мини,* обеспечивающие подключение контроллеров ПУ, устройств оперативной и внешней памяти, устройств межсистемной связи, спецпроцессоров; построение низовых локально-распределенных МПС с мощной системой ввода-вывода (СВВ); подключение серийно выпускаемых УСО различного типа, в первую очередь для создания микроИВК; наладку, контроль, испытание и промышленный выпуск сложных контроллеров ВЗУ; дисплейных процессоров, САМАС-процессоров; построение дешевых инструментальных систем для разработки проблемно-ориентированного программного обеспечения для разрабатываемого периферийного оборудования мини-ЭВМ.

Основное назначение *гомогенных УСИ-микро* — создание инструментальных систем для разработки и документирования на мини-ЭВМ программного обеспечения для систем на основе микроЭВМ; подключение уникальных устройств спецпроцессоров микроЭВМ, построенных на основе последних достижений микроэлектроники (в первую очередь ОЗУ большой емкости, матричных процессоров, процессоров языков высокого уровня и т. д.); экономия оборудования при подключении значительного количества периферийных устройств (за счет использования наборов интерфейсных БИС, разработанных для магистрали микроЭВМ).

Гетерогенные УСИ предназначены для подключения устройств массовых ЭВМ и спецпроцессоров к более мощным новым микропроцессорам [4, 5].

Рассмотрим основные решения в части согласования интерфейсов, реализованные в системах, построенных на основе совместимых семейств мини- и микроЭВМ различных архитектур.

Общая характеристика системных интерфейсов

Системные интерфейсы предназначены для объединения блоков ЭВМ в единую систему. Тенденции их развития определяются необходимостью существенного увеличения операций ввода-вывода, номенклатуры и числа ПУ. Рост удельного объема интерфейсного оборудования в составе ЭВМ обусловливает ужесточение требований на унификацию и стандартизацию интерфейсов.

К характерным особенностям системных интерфейсов относятся возможность функционирования в нескольких режимах взаимодействия; ввод-вывод по программному каналу и по каналу прямого доступа (КПД) в память, что накладывает соответствующие условия на функциональный состав систем шин. На КПД возглавляются функции разгрузки процессора по передаче информации, а его настройка и обработка прерываний от ПУ осуществляется под программным управлением процессора. Появление КПД в составе ЭВМ — результат компромисса между требованием простоты схемного оборудования и желанием повысить производительность ЭВМ за счет совершенствования СВВ. Совершенствование технологии сделало экономически обоснованным применение в составе ЭВМ микропрограммируемых КПД на базе МП. К особенностям системных интерфейсов, оказывающим влияние на их функциональную организацию и состав шин, относится также необходимость сопряжения модулей ОЗУ и ПУ; для этого вводится ряд дополнительных сигналов, обеспечивающих специфические режимы функционирования модулей ОЗУ.

Интерфейсы микроЭВМ имеют определенные отличия от интерфейсов мини-ЭВМ, связанные прежде всего с ограничениями функционального и конструктивного характера. К ним относятся необходимость минимизации внешних выводов БИС, низкая мощность выходных сигналов, а также упрощение и удешевление самой ЭВМ. Благодаря высокому уровню интеграции и универсальности блоков микроЭВМ обеспечивается их минимальная номенклатура. Как правило, это МП, ОЗУ, ПЗУ и модули ввода-вывода со встроенной интерфейсной частью. В структуре связей микроЭВМ формируются *внутренний интерфейс*, объединяющий БИС процессора, ОЗУ, ПЗУ, управления вводом-выводом и *внешний интерфейс*, обеспечивающий сопряжение между внутренней шиной и ПУ. В настоящее время наблюдается тенденция к замене двухуровневого системного интерфейса с раздельными шинами адреса и данных, характерного для ЭВМ 70-х годов, на интерфейс с минимальным числом шин на основе использования принципа разделения времени (мультиплексирования). Часто совмещаются линии адреса и данных, сокращаются шины приоритетной выборки и упрощаются процедуры селекции.

Основными показателями технического уровня СВВ и системных интерфейсов служат: тип и число КПД, возможность непосредственного обмена данными между ПУ, минуя процессор; число уровней приоритета; возможности системы адресации; максимальная скорость передачи при максимальной длине связей; время идентификации источника запроса; число линий в системе шин. Выбор варианта системного интерфейса микроЭВМ — всегда компромисс между простотой реализации и возможностью обеспечения максимальной пропускной способности и производительности СВВ. В качестве признака, характеризующего структуру связей в ЭВМ и МПС, можно использовать степень совмещения информационных магистралей: внутренней информационной шины процессора, ОЗУ, ввода-вывода. По данному признаку ЭВМ и МПС можно разделить на однопроцессорные ЭВМ с единой магистралью и с раздельными магистралями ОЗУ и ввода-вывода, магистрально-модульные МПС.

Системные интерфейсы можно разделить по совокупности значений двух показателей: времени взаимодействия T и максимальному расстоянию взаимодействия между составными элементами интерфейса на шине L. При этом T определяется как время доступа процессора к ПУ, необходимое для пересылки единицы информации, L — максимально возможная длина магистрали. В соответствии с T, L системные интерфейсы разделяются на ряд основных групп (табл. 1).

Такая классификация позволяет дать общую картину функциональной ориентации интерфейсов по областям применения. Указанные границы T и L могут быть расширены при использовании расширителей и ретрансляторов, обеспечивающих удлинение системной магистрали и увеличивающих нагрузочную способность при некотором уменьшении физической скорости передачи. При построении ретрансляторов используют способ непосредственной параллельной передачи всех сигналов магистрали, либо выполняют дополнительное преобразование в последовательный код с последующим последовательно-параллельным преобразованием сигналов на приемном конце.

Исходя из общей характеристики системных интерфейсов, данной в этом разделе, можно заключить, что основная тенденция их развития связана с ориентацией на универсальность в использовании, обеспечивающей широкие функциональные возможности построения МПС. Повышение уровня стандартизации интерфейсов приведет к увеличению объема массового производства и сферы применения ЭВМ.

Интерфейсы ЭВМ с единой магистралью

Данная группа интерфейсов достаточно хорошо освещена в литературе. Поэтому при их рассмотрении акцентируем внимание только на характерных особенностях и отличиях при конкретной реализации в ЭВМ различных классов [3, 6]. Эти интерфейсы представляют собой совокупность совместимых и частично совместимых магистралей, предназначенных для рационального построения ЭВМ различных классов (табл. 2). Основу всех модификаций интерфейсов данного класса составила система сопряжения Unibus. Аналогичный интерфейс «Общая шина» (ОШ) является системным интерфейсом СМ ЭВМ.

Совмещение информационных магистралей ОЗУ и ввода-вывода позволило максимально унифицировать внутренние связи ЭВМ, внесло качественно новые особенности в архитектуру и в системотехнические возможности использования ЭВМ. Общая система адресации ОЗУ и ПУ теоретически позволяет подключать значительно большее число ПУ, чем

Таблица 1

Классификация системных интерфейсов

Группа	T, мкс	L, м	Наименование интерфейса
Внутриплатные	1	0,3	Microbus, Интербис; модификация Q-bus; МПИ
Сосредоточенные	1	0,8	Q-bus персональных ЭВМ; магистраль Nova; Multibus I, II; И41 СМ
Локально-сосредоточенные	5	3	Q-bus, МПИ микроЭВМ; И41 СМ
Локальные	10	15	Unibus PDP-11, ОШ СМ
Локально-распределенные	15 и более	20 и более	Последовательная магистраль Micro-Nova

в ЭВМ двухуровневой структуры, обрабатывать содержимое регистров ПУ с помощью адресных команд обращения в память, исключая необходимость в специальных командах ввода-вывода. Единая система синхронизации асинхронного типа и возможность адресации ОЗУ со стороны ПУ позволяет подключать к процессору модули ОЗУ, ПУ различного быстродействия и выполнять обмен информацией между ними по КПД с совмещением выполнения операций в процессоре.

Реализация магистрали на основе приемопередатчиков (ПП) с открытым коллектором и кабельных соединений (в пределах 20 м) частично решает проблему расстояния и позволяет создавать двух и многомашинные системы, строить системы, имеющие широкие функциональные возможности по резервированию ЭВМ и коллективному использованию несколькими процессорами наиболее дорогостоящих ПУ.

К недостаткам данных системных интерфейсов можно отнести:

— необходимость динамического перераспределения информационной магистрали между процессором и ПУ, что повышает общее время установления связи и время реакции на прерывание;

— увеличение длины и унификация связей интерфейса снижает надежность системы в целом, так как любое короткое замыкание или обрыв одной из линий вызывает отказ всей системы;

— реализация модулей характеризуется повышенным объемом их интерфейсного оборудования.

Интерфейс ОШ в ранних 16-разрядных мини-ЭВМ и в современных 16-разрядных многоплатных микроЭВМ используется в качестве унифицированной системы связей и сигналов между одним процессором, ОЗУ, контроллерами ВЗУ, ПУ из номенклатуры устройств системы. Интерфейс реализуется с использованием магистральных ПП средней степени интеграции, плоских кабелей и соответствующих разъемов. В интерфейсе предусмотрены средства защиты информации при отказах сетевого питания.

Технически магистраль выполняется в двух основных модификациях. Внутри каркасов, содержащих функционально законченные устройства (ЦП, ОЗУ, контроллеры системных ПУ) она реализуется в виде соединения разъемов (иногда магистрали разъемов) с ограничениями на длину отводов. Каркасы внутри основной стойки ЭВМ и контроллеры ПУ, локально-расположенные вне стойки, соединяются между собой кабелем. Для сокращения времени цикла при работе с ОЗУ, располагаемым рядом с блоком процессора, в магистрали более производительных мини-ЭВМ (типа СМ4) используют ускоренную процедуру асинхронного взаимодействия путем сокращения времени, отводимого для компенсации разброса параметров ПП.

Таблица 2

Характеристики интерфейсов ЭВМ с единой магистралью

Параметр	Мини-ЭВМ	МикроЭВМ	Персональные ЭВМ
Число функциональных линий (Ст/Доп)	56/60	34/46	46
Разрядность данных, бит	16	16	16
Разрядность шин адреса, (Ст/Доп)	18/22	18/22, М	22, М
Число линий управления	22	16	24
Число уровней приоритета, (Ст/Доп)	5	2/5	3
Арбитраж приоритетов ПУ, (ЦЦ, ЦР)	ЦЦ	ЦЦ	ЦР
Управление режимом ПДП, (Ц)	Распределенное	Распределенное	Централизованное
Система адресации ПУ, (Л, Г)	Логическая	Логическая	Г+Л
Идентификация наличия модуля	Отсутствует	Отсутствует	Имеется
Число подключаемых устройств	18	15	8
Длина магистрали, м	20	1	0,5
Минимальное время цикла, мкс	0,4	0,5	0,5

Примечание: Ст — стандартное, Доп — допустимое, М — мультиплексируемые, Л — логическая, Г — географическая, ЦЦ — централизованный цепочечный, ЦР — централизованный радиальный, Ц — централизованное.

В мини-ЭВМ, обеспечивающих расширенную адресацию ОЗУ (до 22-х разрядов), для связи с ОЗУ применяют специализированную магистраль, совместимую с основной. В этом случае ОШ является магистралью ввода-вывода для всех типов периферийного оборудования. В высокопроизводительных ЭВМ ОШ служит магистралью ввода-вывода для ПУ малой и средней производительности. Для подключения контроллеров высокопроизводительных ВЗУ используют специализированную магистраль, содержащую 32-разрядную шину данных, связанную через соответствующий адаптер с ОЗУ. Указанные особенности реализации системных магистралей мини-ЭВМ препятствуют их стандартизации на уровне национальных стандартов. Существенным ограничением на более широкое применение является отсутствие в системных интерфейсах средств обеспечения режима мультипроцессорности. Это в такой же степени относится к магистралям типа ОШ микро- и персональных ЭВМ.

В системном интерфейсе микроЭВМ используется мультиплексирование шин адреса и данных в целях реализации магистрали в БИС с корпусом, имеющим 40 выводов. В магистрали массовых микроЭВМ 16 двунаправленных линий мультиплексируются во времени относительно адресов и данных. Передачи по ним упорядочиваются при помощи управляющих линий, обеспечивающих идентификацию и направление обмена по информационным шинам, различение передачи слова и байта данных, синхронизацию обмена.

Для упрощения СВВ используется специальный сигнал, идентифицирующий область адресов ПУ. Одноуровневая система управления прерываниями и операциями ПДП построена по цепочечной схеме. Три сигнала необходимы для установки системы в исходное состояние при включении (выключении) питания. Дополнительные три линии выполняют системные функции: прекращение эмуляции набора команд и переход в режим работы с пультом перезаписи микрокода в динамической памяти; операции в реальном масштабе времени.

Развитие системного интерфейса связано с созданием микроЭВМ на базе современных МП, сравнимых по возможностям и производительности с процессорами мини-ЭВМ. Это потребовало увеличения разрядности адресных линий до 22-х и использования более развитой подшины управления прерываниями, применяемой в ОШ.

При выборе системного интерфейса и ПУ персональных ЭВМ принимают во внимание главным образом требования максимальной простоты подключения и отключения ПУ при минимальной стоимости, получаемой за

счет отказа от совместимости с соответствующими контроллерами ПУ микроЭВМ [7]. Основная панель персональных ЭВМ моделей 325 и 350 фирмы DEC реализует новые принципы конструктивной простоты выполнения интерфейса, обеспечивающие возможности установки отдельных модулей в любое посадочное место.

Магистраль рассчитана на применение недорогих ПП с тремя состояниями (восемь в корпусе), что обеспечивает более экономичное подключение к шине и улучшение электрических параметров линий шины при ограничении числа посадочных мест до восьми. Специальные средства доступа к шине исключают конфликтные ситуации между двумя одновременно работающими передатчиками. Использование стробирующих сигналов «разрешение задатчика» и «разрешение исполнителя» позволяют задатчику управлять работой ПП таким образом, чтобы информация передавалась от исполнителя только в нужном направлении по адресу, указанному задатчиком. Это исключает возможность потерь данных, обусловленных импульсными помехами от источника питания в моменты переключения ПП.

Синхронизация операций ПДП осуществляется системным модулем. Базовые функциональные компоненты (ЦП, ОЗУ емкостью 256 Кбайт; ПЗУ, содержащее программы диагностики и инициализации; энергонезависимое ОЗУ с синхронизатором; контроллеры АЦПУ, асинхронная и синхронная линии связи) подключаются к внутренней секции интерфейса. Компоненты расширения моделей (массовая сменная и дополнительная память на ГМД и дисках типа Винчестер и другие модули) подсоединяются к дополнительной секции интерфейса, связанной с основной через соответствующий буфер. Подключение основных контроллеров ПУ и ОЗУ к внутренней секции интерфейса позволяет реализовать более высокие скорости передачи данных и обеспечивает экономию технических средств и физического объема.

Три уровня приоритета (2, 1, 0 — в порядке убывания) осуществляют управление интерфейсом. Устройство более высокого приоритета, имеющее запрос на обращение к интерфейсу, получает управление шиной только по завершении передачи данных устройством с наивысшим текущим приоритетом. Приоритеты прерываний и ПДП не зависят от положения модулей в основной панели, что упрощает подключение кабелей ПУ и обеспечивает дополнительную гибкость.

Используемая система назначения фиксированных адресов посадочным местам (географическая адресация) также является дополнительным преимуществом. Для каждого из восьми физических гнезд резервируется один 128-байтовый сегмент страницы ввода-вывода. Дешифратор адресов располагается на основной плате модели и вырабатывает сигнал выборки модуля в адресной части цикла. Младшие семь разрядов адреса дешифрируются в модуле для идентификации одного из 128 байтов, к которому производится обращение.

Каждый дополнительный модуль идентифицируется индивидуальным сигналом, посылаемым в системный модуль при включении питания, и фиксируется в виде бита в соответствующей ячейке памяти страницы ввода-вывода. Диагностическая программа, хранящаяся в ПЗУ системного модуля, анализирует значения разрядов этого регистра, определяя размещение дополнительных модулей. При включении питания предусмотрена полная проверка целостности структуры, работоспособности всех функциональных узлов модели и установленных дополнительных модулей. В случае обнаружения ошибок выдаются сообщения на терминал и индицируются ошибки на системном модуле.

Интерфейсы ввода-вывода ЭВМ

Для всех моделей мини-ЭВМ с интерфейсами ввода-вывода обычно используется унифицированная параллельная магистраль, обеспечивающая подключение периферийного оборудования, работающего по программному каналу и КПД [8]. Структура магистрали Nova фирмы Data General довольно типична для магистралей мини-ЭВМ общего назначения, однако отличается от них специальными режимами работы (инкрементным и накопления). В инкрементном режиме по указанному контроллером ПУ адресу ячейки ОЗУ ее содержимое увеличивается на единицу. В режиме накопления данных по указанному контроллером ПУ адресу ячейки ОЗУ ее содержимое суммируется с 16-разрядными данными ПУ, а результат передается ПУ в виде 16-разрядного слова. Специальные режимы работы магистрали обеспечивают более эффективное использование мини-ЭВМ в системах автоматизации научных и научно-технических экспериментов, позволяя непосредственно получать в ОЗУ гистограммы работы ПУ.

Магистраль ввода-вывода микроЭВМ семейства уникальна для большинства микроЭВМ общего назначения. МикроЭВМ использует 16-разрядный МП с раздельными шинами адресов и ввода-вывода. Последние содержат шесть линий, с помощью которых протокол параллельной магистрали реализуется в последовательном коде. Приемопередающие БИС, расположенные в процессоре и контроллерах ПУ, осуществляют параллельно-последовательное преобразование протокола последовательной магистрали в протокол параллельной магистрали ввода-вывода за исключением специальных режимов передачи данных по КПД.

Основные ПУ подключаются к последовательной магистрали непосредственно, дополнительные — посредством интерфейсной карты общего назначения. Карта содержит БИС, являющуюся предпроцессором шин ввода-вывода МП и имеет модификации, обеспечивающие подключение необходимых ПУ и согласование с магистралями ЭВМ других систем. Контроллер выполняет шифрацию команд ввода-вывода, формирует запросы прерывания, управляет КПД.

Интерфейсы 8-, 16-разрядных МПС

По типу системных интерфейсов, МПС разделяются на два класса, использующие в качестве магистрали ввода-вывода внутреннюю магистраль МП, доработанную до унифицированного канала ввода-вывода, или внешнюю по отношению к МП системную магистраль. В основу большинства 8-разрядных МПС первого класса положены магистрали МП типа Intel 8080 и Motorola 6800 (табл. 3). Эти семейства имеют высокоразвитые процессоры ввода-вывода в виде БИС, позволяющие экономично реализовывать СВВ различной сложности. Магистрали используют пространственное разделение шин адреса и данных, что позволяет существенно увеличить быстродействие МПС. Усилители шин полностью совместимы с ТТЛ-микросхемами. Шина управления обеспечивает синхронизацию работы МП и магистрали ввода-вывода, внешней памяти, ПУ, управление прерыванием, режимами ПДП, выдачу в магистраль состояния процессора и т. д.

Стремление к стандартизации 8-разрядных МПС обусловило широкое распространение в США магистрали MUBUS, гарантирующей совместимость с большинством массовых 8-разрядных МП (Intel 8085, Motorola 6800, TMS-9900, Z-80 и др.). Набор сигналов близок к Microbus Intel (16 адресных, 8 данных и 10 управления) и обеспечивает расширение и создание МПС с большим числом ПУ.

В большинстве 16-разрядных МПС используются процессорно-независимые магистрали, ориентированные на создание мультимикропроцессорных систем, размещенных, как правило, в одном каркасе, рассчитанным на установку 2—10 модулей. Наиболее известными среди них и широко применяемыми являются магистрали ти-

Таблица 3

Характеристики магистралей 8-разрядных МПС

Характеристика	Intel 8080	Motorola 6800
Разделение информационных шин	Пространственное	
Разрядность шин адреса	16	16
Разрядность шин данных	8	8
Число линий управления	10	5
Общее число линий (с резервными)	34	30
Операции прямого доступа	Отсутствуют	
Тип арбитража	Цепочечный	
Контроль состояния магистрали	Отсутствует	
Число подключаемых каркасов	1	1
Реализация усилителей магистрали	С тремя состояниями	
Число подключаемых устройств	8	10
Максимальное расстояние, м	0,6	1,0
Максимальное быстродействие, МГц	0,5	0,5

па S-100 (стандарт IEEE P-696), Multibus (варианты стандартов IEEE P-796 и AMS-bus), Versabus фирмы Motorola с пространственным разделением шин адреса и данных, позволяющие строить МПС с высокоскоростной мультипроцессорной магистралью [9, 10].

В некоторых 16-разрядных МПС, ориентированных на использование МП отличных от LSI-11 и периферийного оборудования микроЭВМ фирмы DEC, применяется в качестве системного интерфейса Q-bus. В этом случае в центральной плате МПС размещаются средства сопряжения системных интерфейсов [4]. Для обеспечения совместимости 8- и 16-разрядных МПС в этих магистралях предусматриваются соответствующие режимы работы с 8-разрядными устройствами.

Методы и этапы проектирования УСИ

Основные решаемые задачи. На основе формальных моделей системных интерфейсов необходимо выработать единые принципы классификации, описания структурной и функциональной организации УСИ; используя известные методы, предложить методы оценки эффективности реализации УСИ различной архитектуры. Последовательность системного этапа проектирования структуры УСИ состоит из процедур рассмотрения возможных и определения допустимых альтернативных вариантов организации УСИ; выбора критерия эффективности технического уровня структуры УСИ; выявления наиболее целесообразного для МПС варианта УСИ на подмножестве допустимых альтернативных вариантов в условиях заданных ограничений. В качестве критерия эффективности используются показатели практической оптимальности, используемые при проектировании аналогичных устройств [11]. Результатом системного проектирования служит набор функций, структура УСИ и способы реализации интерфейсных функций.

Технический уровень УСИ характеризуют следующие ключевые принципы: режим передачи по программному каналу с типом коммутации уровней прерывания (гомогенные, гетерогенные); режим передачи по ПДП (мультиплексный, монопольный); дополнительные системные функции (расширение адресации, совмещение работы сопрягаемых интерфейсов, наличие КПД общего назначения и т. д.); техническая реализация (способ организации управления, тип используемых ПП и ИМС и т. п.); тип структуры СВВ с УСИ, определяющий структуру и в основном конструктивное исполнение УСИ. Последние две функции являются специфичными для универсальных УСИ-мини.

Возможны следующие основные типы УСИ-мини, определяющие системную организацию подключаемых к УСИ устройств: эмулирующие только программный канал мини-ЭВМ; универсальные, эмулирующие полностью системный интерфейс; универсальные с расширенными системными возможностями; специальные, обеспечивающие наиболее эффективную работу СВВ в целом (табл. 4).

Устройства первого типа наиболее просты, реализуются в виде платы микроЭВМ и устанавливаются в основную ее кассету. Разрабатываются на начальном этапе создания систем, использующих устройства низкого и среднего быстродействия. Универсальные эмулирующие УСИ получили широкое применение в МПС на базе LSI-11/02, LSI-11/23 и т. д. для подключения любого оборудования мини-ЭВМ за исключением процессора. Наибольшая эффективность обеспечивается при использовании микроЭВМ с УСИ вместо процессора мини-ЭВМ меньшей производительности и (или) надежности. Универсальные УСИ с расширенными возможностями ориентированы на гибкие МПС на базе мощных МП с повышенными требованиями к объему ОЗУ и динамическому изменению приоритетов запросов. Основной недостаток — необходимость модернизации операционных систем микроЭВМ. Специальные УСИ являются альтернативным, более быстродействующим вариантом УСИ третьего типа и в основном используются в МПС с распределенным управлением.

Достоинства МПС с УСИ (табл. 5) можно качественно оценить на примере типовой системы, использующей МП с ОЗУ, ПУ базового комплекта микроЭВМ и соответству-

Таблица 4

Характеристики организации УСИ-мини

Характеристика	Тип УСИ-мини				
	1	2	3	4.1	4.2
Передача по программному каналу	+	+	+	+	+
Число коммутируемых линий прерывания	1, 2, 4	2,4	4	4	4
Передача по прямому доступу:	—	+	+	+	+
монопольный режим	—	+	+	—	+
мультиплексный режим	—	+	+	+	+
Техническая реализация:	+	+	+	+	+
схемная	+	+	+	+	+
микропрограммная	—	—	+	+	+
Дополнительные системные функции:					
расширение адресации	—	—	+	+	—
совмещение работы интерфейсов	—	—	—	+	+
встроенный КПД	—	—	—	—	+
Тип структуры СВВ с УСИ:					
локальная	+	+	+	+	—
локально-распределенная	—	—	—	—	+

Таблица 5

Сравнительная оценка МПС с УСИ-мини

Характеристика	Тип УСИ				
	1	2	3	4.1	4.2
Расширение объема ОЗУ	0	1	2	2	0
Производительность СВВ	0	1	1	2	2
Система прерывания СВВ	1	1	2	1	1
Разделение во времени работы на интерфейсе	0	1	1	2	2
Мультисистемные возможности	0	1	1	1	2

Примечание: 0 — обычные возможности, 1 — улучшенные, 2 — наилучшие.

ющего УСИ-мини при помощи известных методов [12]. Тенденции развития УСИ связаны с появлением новых более мощных МП, специальных ПУ и спецпроцессоров, характеризующихся потенциально снижающейся стоимостью. Число потребителей МПС с УСИ быстро растет, а актуальные области их применения расширяются. К особенностям проектирования УСИ можно отнести изменение конструктивных и логических требований системных интерфейсов и появление новых стандартных мультипроцессорных системных интерфейсов, что приводит к поиску более рациональных структур УСИ, необходимости предотвращения больших затрат технических средств и времени на синхронизацию работы процессора с устройствами, выходящими на другой системный интерфейс.

Развитие архитектуры и структуры УСИ определяется не только улучшением характеристик БИС, МП и микроЭВМ, но и тем, что МПС с УСИ могут более широко использоваться в качестве компонентов МПС с различными видами сложных структур СВВ и многофункциональных подсистем связи с процессами и объектами. На этот процесс оказывает существенное влияние разработка и реализация принципов и развитие малых локально-распределенных ВС с УСИ, использующих магистрально-последовательные принципы соединения компонентов СВВ [13, 14]. Эта тенденция получает все более широкое развитие с появлением коммерческих МП со встроенными последовательными магистралями, БИС последовательных магистралей обмена, оптоволоконных линий связи.

ЛИТЕРАТУРА

1. Мячев А. А., Никольский О. А. Программно-модульные цифровые системы на базе машинно-независимых интерфейсов. Обзорная информация, вып. 5, ТС-2: ЦНИИТЭИприборостроения, 1983. — 46 с.

2. Пролейко В. М. Микропроцессорные средства вычислительной техники и их применение. — Микропроцессорные средства и системы, 1984, № 1, с. 11— 16.

3. Мячев А. А. Системы ввода-вывода ЭВМ. — М.: Энергоиздат, 1983. — 168 с

4. Мячев А. А., Никольский О. А. Стандартные интерфейсы микропроцессорных систем. — Микропроцессорные средства и системы, 1984, № 1, с. 27—33.

5. Мячев А. А., Снегирев А. А., Володарский М. И. Вариант согласования интерфейсов СМ1800 и СМ4. В кн.: Автоматизация научных исследований на основе применения ЭВМ. Новосибирск, ИАиЭ, 1981, с. 59—60.

6. Эккауз Р., Моррис Л. Мини-ЭВМ: организация и программирование. — М.: Финансы и статистика, 1983. — 359 с.

7. Электроника, 1982, № 15, с. 38—46.

8. Computer. — Datapro Research Corp., May, 1983.

9. EDN, May, 1981, p. 159—173.

10. EDN, June, 1981, p. 146—186.

11. Мячев А. А. Об одном методе оценки технического уровня контроллеров крейта КАМАК. — Автометрия, 1981, № 6, с. 99—102.

12. Вейцман К. Распределенные системы мини- и микроЭВМ. — М.: Финансы и статистика, 1983. — 382 с.

13. Мячев А. А. Кольцевая система с последовательной передачей сигналов общей шины. — Управляющие системы и машины. — Киев: Наукова думка, 1978, № 2, с. 86—88.

14. Hewlett-Packard Jornal. Jan., 1983, p. 3—22.

Статья поступила 20 февраля 1984 г.

РЖ ВИНИТИ

Операционные системы для 16-разрядных микропроцессоров. Options abound in 16-bit operating software. «Comput. Des.», 1983, 22, № 9 (англ.)

Анализируются различные подходы к проблеме создания оптимальной операционной системы для 16-разрядных МП. Даются краткие характеристики систем, разработанных американскими фирмами для МП типов 68000, 8086, 8088, Z8000. В качестве примера анализируется система типа MS-DOS2.O фирмы MICROSOFT, которая предназначена для МП модели 8088 и ориентирована на работу с одним пользователем. В состав операционной системы включены программы управления терминалами различных типов, что позволяет легко адаптировать систему для работы с любой стандартной периферией.

Система позволяет реконфигурацию вводов и выводов, давая возможность программировать число входов и выходов в зависимости от потребности. В системе используется динамический доступ к секторам дисковой памяти, что повышает скорость, в частности при работе с БД. Операционная система дает возможность выполнять, кроме основной, также и фоновые задачи, используя для этого времени прерывания МП.

Реализация операционных систем реального времени на встроенных микроЭВМ. Implementierung von Echtzeitbetriebssystemen fur eingebettete Mikrorechner. Werner D. «Weiterbildungszentr. math. Kybern. und Rechentechn./Informationsverarb. Dresden», 1982, № 63 (нем.)

Обычные операционные системы (ОПС) редко применяются для микроЭВМ. Вместо этого в потребительских программах реализуются отдельные функции ОПС. Сообщается о разработке набора функциональных модулей для ОПС реального времени. Пользователь имеет возможность создавать ОПС, которые оптимальным образом подходят для конкретного применения. Количества задач, служебных функций, внешних событий, сообщений и т. п. задаются управляющими таблицами. Выбираются алгоритмы организации очередей и системные функции.

Для реализации дополнительных функций пользователь сам может разработать функциональный модуль. В максимальном варианте ОПС занимает 1800 байт в ПЗУ. Управляющие таблицы для 20 задач и 40 служебных функций занимают 500 байт в ОЗУ. Время выполнения системных функций (при коротких очередях) занимает от 0,1 до 0,4 мс. [Techn. Univ. Dresden, ГДР].

УДК 681.3—181.48

В. М. Брябрин

ИНТЕЛЛЕКТУАЛЬНЫЙ ИНТЕРФЕЙС НА ОСНОВЕ ПЕРСОНАЛЬНОЙ ЭВМ

Реализация удобных программных средств для взаимодействия пользователей с прикладными системами на персональных ЭВМ позволяет создавать автоматизированные рабочие места, отличающиеся компактностью, невысокой стоимостью, надежностью, удобством в работе и гибкими функциональными возможностями.

Автоматизированные системы, создаваемые на основе больших и мини-ЭВМ, функционируют, как правило, в операционной обстановке, обусловленной общей организацией работ в вычислительных центрах коллективного пользования. Эта обстановка задает строгие правила входа в систему и проведения сеансов, требует знания особенностей машин и их программного обеспечения и в целом предполагает, что человек, работающий на машине, должен обладать определенной программистской подготовкой. Вследствие этого многие потенциальные пользователи ЭВМ — научные работники, проектировщики, экономисты, инженерно-технический персонал, медики, педагоги — отделены от вычислительных средств своеобразным барьером «некоммуникабельности» и обращаются к ним лишь через посредство профессиональных программистов.

Совершенно иначе обстоит дело при использовании нового класса вычислительных машин, получивших название персональных ЭВМ (ПЭВМ). Современные ПЭВМ дают большие возможности для упрощения человеко-машинного взаимодействия и его существенного обогащения по сравнению с большими и малыми ЭВМ. Этому способствует возможность индивидуальной настройки ПЭВМ и их автономного функционирования, не зависящего от общей организации ВЦКП. Большое значение имеет простота и «дружественность» операционных систем, *возможность активной работы с текстовой и графической информацией*, а также возможность применения специальных устройств ввода (манипуляторов типа «мышь», «джойстик», графических планшетов и других). Наконец, надежность, компактность, малое потребление энергии и низкая стоимость являются важнейшими факторами, способствующими широкому распространению ПЭВМ и созданию на их основе автоматизированных систем разного назначения.

В данной работе рассматривается метод организации интеллектуального интерфейса на основе ПЭВМ как главного средства взаимодействия человека и автоматизированной системы. При этом ПЭВМ может служить инструментом решения прикладных задач или играть роль промежуточного звена между пользователем и более крупной вычислительной системой.

Требования к интеллектуальному интерфейсу

В предлагаемом подходе интеллектуальный интерфейс строится с учетом следующих факторов.

Многообразие форм человеко-машинного взаимодействия. Необходимо охватить как можно более широкий спектр форм обмена информацией между человеком и машиной: на основе командных языков, меню, таблиц, графиков, текстовых документов, графических изображений, акустических последовательностей. При поддержке такого многообразия необходимо сохранять возможность легкого и естественного переключения с одной формы общения на другую.

Единство стиля работы пользователя. Несмотря на многообразие допустимых форм общения, при выборе любой из них человеку должны предоставляться достаточно однородные средства выбора — ввода — запроса необходимой информации. Это должно относиться, в частности, к средствам указания определенных элементов на экране, способам их удаления, комплексации составных объектов из более простых, сохранения объектов на внешних носителях, повторного вызова и воспроизведения.

Интеграция представления данных. Каждая форма представления информации основывается, как правило, на своем, наиболее эффективном способе реализации. Задача построения интеллектуального интерфейса предполагает интеграцию различных методов представления информации. Это может осуществляться на основе единого внутреннего представления данных либо путем автоматического преобразования одних представлений в другие.

Унификация связей с прикладными системами. Программные средства, реализующие различные формы взаимодействия и средства представления-преобразования объектов, должны быть оформлены как *унифицированные программные модули*, связь которых с прикладными системами осуществляется по общим правилам взаимодействия модулей, в рамках определенного языка программирования или операционной системы ПЭВМ.

Таким образом, интеллектуальный интерфейс представляет собой внешнюю оболочку, отделяющую пользователя от остального программного и информационного обеспечения ПЭВМ. При этом человеку дается богатый выбор форм (средств) отображения получаемой информации и ввода своих команд, запросов и указаний с соблюдением единого стиля работы. Одновременно регламентируется способ связи между компонентами интеллектуального интерфейса и прикладными системами. При проектировании интеллектуального интерфейса необходимо уделять внимание двум взаимосвязанным компонентам:

средствам поддержки человеко-машинного взаимодействия и *средствам их конструирования.*

Наша цель состоит в том, чтобы средства первой группы по возможности входили во вторую, обеспечивая тем самым максимальную *самоподдержку* системы.

Общая концепция и состав программного обеспечения

Одна из базовых идей организации интеллектуального интерфейса состоит в том, что пользователю предлагается «объектно-ориентированный» взгляд на прикладную автоматизированную систему. Исследуемая область представляется при этом *множеством проблемно-ориентированных объектов* {B}, каждый из которых соответствует абстрактному понятию, классу физических объектов, процессу, состоянию или их конкретным воплощениям — экземплярам. Объекты {B} находятся в определенных взаимоотношениях, которые выражаются законами {R}, принимающими форму логических высказываний, функциональных зависимостей или обычных программ. В основе такого подхода лежит опыт реализации диалоговой информационно-логической системы ДИЛОС [1], а также опыт создания инструментальных систем для реализации пакетов прикладных программ — ПРИЗ [2], СПОРА [3], МАВР [4].

В соответствии с развиваемым подходом, общение пользователя с системой должно осуществляться в рамках сценария, который формально задается множеством специальных объектов {D}, образующих *схемы диалога*. Объекты могут создаваться и подвергаться преобразованиям со стороны прикладных программ или по инициативе пользователей. Благодаря этому возникает возможность создания адаптивной диалоговой среды, легко настраиваемой на конкретные потребности пользователей [5].

При работе системы человек должен иметь возможность воспринимать и воздействовать на внешние представления всех релевантных объектов — {E}, которые могут принимать следующие основные формы:

E_1 — линейные тексты, E_2 — форматированные текстовые структуры, E_3 — математические формулы, выражения, тексты программ, E_4 — таблицы, E_5 — математические графики, диаграммы, гистограммы, E_6 — условные графические символы и их композиции, E_7 — произвольные графические изображения.

Формы $E_1 \dots E_3$ будем относить к классу текстовых представлений $\{E_t\}$, формы $E_5 \dots E_7$ — к классу графических представлений $\{E_g\}$. Форма E_4 занимает промежуточное положение, так как содержит и графические и текстовые элементы.

В составе интеллектуального интерфейса должны быть программные средства, обеспечивающие отображение внутренних представлений объектов во внешние ($B \rightarrow E$), средства интерактивного воздействия на внешние представления ($E \rightarrow E'$) и отображения результатов этих воздействий во внутренние представления ($E' \rightarrow B'$).

Важно отметить, что элементы множеств {B} и {R} также могут идеть адекватные внешние представления E(B) и E(R), поэтому воздействие на них указанными средствами позволяет легко приспосабливать систему к конкретным потребностям, причем единство стиля работы дает возможность самим потребителям осуществлять необходимую настройку системы, не прибегая к помощи профессиональных программистов.

Таким образом, интеллектуальный интерфейс представляет собой совокупность программных средств для согласованной работы с множествами (B, R, D, E). Программные компоненты интеллектуального интерфейса могут быть реализованы как *библиотеки процедур (функций)*, присоединяемые к прикладным программам на уровне загрузочных или объектных модулей и как *специальные компоненты операционной системы (драйверы)*, доступные прикладным программам в соответствии с соглашениями о связях.

Первый способ более эффективен и целесообразен при использовании одного и того же языка программирования для реализации интеллектуального интерфейса и прикладных программ. Второй — обеспечивает языковую независимость интерфейса и прикладных систем, но требует определенных «накладных» расходов для поддержки соглашений о связях между прикладными программами и операционной системой.

В состав программного обеспечения интеллектуального интерфейса* входит несколько базовых компонентов:

система управления дисплейными окнами (СУОК) — обеспечивает работу с текстовыми представлениями объектов $\{E_t\}$;

графический интерфейс (ГРИН) — содержит средства формирования, модификации и воспроизведения графических представлений объектов $\{E_g\}$;

адаптивная диалоговая среда (АДС) — имеет в своем составе конструктор и интерпретатор схем диалога, представляемых множеством диалоговых состояний {D};

объектная база (ОБ) — совокупность средств, обеспечивающих формирование и использование проблемно-ориентированных объектов {B} и связывающих их законов {R} при интерактивном взаимодействии пользователей с прикладными системами.

Ниже излагаются основные принципы реализации указанных компонентов, а на цветном развороте приведены фрагменты работы прикладных программ на персональных ЭВМ.

Система управления дисплейными окнами. При работе с текстовыми представлениями объектов пользователю может быть предложен единый стиль общения с системой, основанный на метафоре текстового редактирования. В разработанной А. Б. Борковским системе СУОК вводится абстрактный тип window (окно), обладающий рядом статических параметров и ассоциированных процедур, которые обеспечивают необходимые манипуляции с самим окном и воспроизводимым в нем текстовым представлением объекта.

```
TYPE window=RECORD
     line ∅, line1,...: line;                      {геометрические параметры окна}
     col ∅, col1,...: column;
     default-color,...: char;                      {цвет фона и другие атрибуты}
     mode: (insert, overtype, read-only);          {режим редактирования строк}
END;
PROCEDURE Init-window (...);                       {создание окна}
PROCEDURE Open-window (VAR W: window);             {открытие окна}
PROCEDURE W-insert (before: line; n: INTEGER);     {вставка строк}
PROCEDURE W-delete (after: line; n: INTEGER);      {удаление строк}
PROCEDURE W-update (L: line; new-line: string);    {обновление строки}
```

Прикладная программа, взаимодействующая с СУОК (рис. 1), пользуется типом window для текстового представления своих объектов E_t (B). В свою очередь, СУОК обращается к процедурам прикладной программы $\{P_t\}$ с целью поддержания соответствия

* Описывается программное обеспечение, разработанное в вычислительном центре АН СССР.

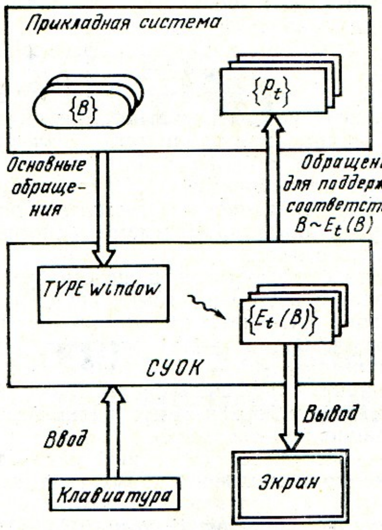


Рис. 1. Взаимодействие системы управления дисплейными окнами (СУОК) и прикладной системы

между объектами В и их внешними, «оконными» представлениями E_t (В). В набор $\{P_t\}$ входят, в частности, такие процедуры: <сдвинуть окно по тексту>, <вставить элемент>, <удалить элемент>, <обновить текст>, <обновить изображение в окне>, <выдать подсказку>, <выполнить спецдействие>.

При создании окна и его открытии на экране дисплея в заданном месте появляется прямоугольная (закрашенная) область, в которую выводится текстовое представление объекта $E_t(B_i)$, состоящее из отдельных элементов. При этом трактовка понятия «элемент» зависит от вида представления. Так, для представления вида E_1 элементом является строка линейного текста, для E_2 — одно из полей структуры, для E_3 — отдельное математическое выражение или оператор программы, для E_4 — клетка или строка таблицы. СУОК воспринимает команды пользователя, передаваемые путем нажатия отдельных клавиш, и выполняет:

стандартные действия, такие как перемещение окна вдоль текста, вставка, удаление и модификация отдельных элементов, переключение активности с одного окна на другое, вызов подсказки, выход из окна;

нестандартные действия, имеющие определенный смысл для каждого приложения.

Стандартные действия вызываются всегда одними и теми же клавишами и реализуются ассоциированными с ними процедурами, вне зависимости от приложений. *Нестандартные действия* определяются функцией <выполнить спецдействие>, аргументом которой является код нажатой пользователем клавиши.

Управление со стороны пользователя может осуществляться не только через клавиатуру, но и через манипуляторы типа «джойстик» или «мышь». При этом сдвиг окна вдоль текста и перемещение курсора в поле окна вызывается непрерывным движением органа управления, а операции удаления, вставки, переключения с одного окна на другое вызываются нажатиями специальных кнопок на манипуляторах.

СУОК позволяет легко реализовывать прикладные системы, в которых представление E_t (В) является естественной формой отображения информации для пользователя. К этой категории относятся системы обработки текстовой документации, редакторы программ, системы табличного ввода данных. Через СУОК может осуществляться просмотр и модификация структурных объектов, имеющих внешнее текстовое представление, и др.

Графический интерфейс. Подобно тому, как СУОК берет на себя функции формирования и отображения текстовой информации, графический интерфейс (подсистема ГРИН) должен поддерживать формирование графических изображений и их использование при решении прикладных задач.

Как указывалось, следует различать несколько основных форм графического представления объектов $\{E_g\}$. Каждой из этих форм соответствует, вообще говоря, свой набор типов данных и обрабатывающих процедур, но ряд системных понятий является общим для всех типов графических объектов (Г-объектов). Определим некоторые из этих понятий как абстрактные типы, состоящие из структур данных и процедур. Основным носителем изображения является структура типа picture (картина).

```
TYPE picture=RECORD
     name: ALPHA;                    {имя картины}
     screen: scr-mode;               {режим экрана}
     basis: abs-coord;               {абсолютные координаты опорной точки — базиса}
     segms: seg-list;                {список сегментов}
END;
PROCEDURE New-picture (VAR p: picture; ...);            {новая картина}
PROCEDURE Save-picture (p: picture; m: mode);           {сохранить в базе}
PROCEDURE Load-picture (name: ALPHA);                   {загрузить из базы}
PROCEDURE Draw-picture (name: ALPHA); q: scale);        {вывод на экран}
PROCEDURE Plot-picture (name: ALPHA); q: scale);        {вывод на плоттер}
PROCEDURE Switch-layer (p: picture; l: layer; f: boolean); {включение-выключение слоя}
```

Картина — это совокупность сегментов (подкартин), размещаемых на определенных позициях относительно фиксированной опорной точки — базиса. Картина может сохраняться в базе данных, загружаться из нее и воспроизводиться на экране или на графопостроителе (плоттере). При воспроизведении может происходить масштабное преобразование всей картины и входящих в нее сегментов.

Она может быть «многослойной» — для каждого сегмента фиксируется слой, в котором он находится. Слои могут «выключаться», т. е. становиться невидимыми, и вновь включаться. В частном случае картина может иметь лишь один слой и содержать один сегмент. Составная часть картины — сегмент определяется следующим образом:

```
TYRE segment=RECORD
     name: seg-name;                 {имя сегмента}
     default-layer: layer;           {слой картины}
     default-basis: coord;           {относительные координаты базиса}
     default-scale: scale;           {масштаб}
     default-angle: angle;           {угол поворота}
     default-color: char;            {основной цвет точек (линий)}
     primitives: prim-list-pointer;  {указатель на список примитивов}
```

```
END;
PROCEDURE New-segm (VAR s: segment; ...);                 {новый сегмент}
PROCEDURE Change-segm-attr (VAR s: segment; ...);         {изменение атрибутов}
PROCEDURE Move-segm (VAR s: segment; c: coord);           {смещение}
PROCEDURE Scale-segm (VAR s: segment; q: scale);          {масштабирование}
PROCEDURE Rotate-segm (VAR s: segment; a: angle);         {поворот}
PROCEDURE Delete-segm (VAR p: picture; s: seg-name);      {удаление сегмента из картины}
FUNCION Copy-segm (s: segment; c: coord): ∧ segment;      {копирование}
```

В соответствии с рекомендациями графического стандарта *GKS* сегмент определяется как список элементарных графических объектов (примитивов). Сегмент размещается в определенном слое картины и все составляющие его примитивы привязаны к одной опорной точке — базису сегмента, который занимает определенное положение относительно базиса всей картины. Изображение, образованное примитивами одного сегмента, может перемещаться по экрану, поворачиваться на фиксированный угол (в простейшем случае — кратный 90°) и масштабироваться относительно своего базиса. При этом возможны *два метода преобразования — прямой и косвенный.*

При прямом преобразовании, выполняемом процедурами Scale-segm и Rotate-segm, происходит пересчет и изменение параметров примитивов, входящих в сегмент, с тем чтобы получилось новое, измененное изображение. Глобальные атрибуты сегмента default-basis, default-scare и default-angle не изменяются.

При косвенном преобразовании, выполняемом процедурами Change-segm-attr и Move-segm, меняются, наоборот, указанные выше глобальные атрибуты сегмента, а координаты (параметры) примитивов не изменяются. Пересчет параметров примитивов происходит каждый раз при выводе изображения сегмента на экран или графопостроитель. Глобальные атрибуты при этом играют роль коэффициентов для пересчета параметров.

Сегменты могут располагаться в разных слоях картины. Копирование сегмента осуществляет функция Copy-segm, возвращающая ссылку на новый экземпляр сегмента. Примитивы, образующие сегмент, представляются соответствующими абстрактными типами. К ним относятся:

— прямые линии, в том числе, отдельно определяемые горизонтальные, вертикальные и ломаные;

— базовые геометрические фигуры: прямоугольник, треугольник, многоугольник, окружность, эллипс;

— дуги окружностей и эллипсов;

— алфавитно-цифровые и другие, специальные символы — стрелки, математические знаки;

— «краски», которыми можно закрашивать любую замкнутую фигуру, начиная с заданной точки внутри фигуры;

— «стирающая резинка» — прямоугольная область заданного размера и цвета, накрывающая часть изображения, в частности, делающая ее невидимой при совпадении цвета резинки с цветом основного фона.

Каждый примитив имеет свой набор параметров (координаты характерных точек, радиус, цвет), а также свой набор обрабатывающих процедур, доступных как программам графического интерфейса, так и прикладной системе. Например, с типом «линия» связываются процедуры (функции) измерения длины, удлинения-укорочения и поворота; с фигурами связаны процедуры изменения параметров; линии, фигуры и символы могут масштабироваться, поворачиваться и менять цвет; у стирающей резинки меняются размеры и цвет.

Картина, сегмент и примитивы являются универсальными типами, позволяющими создавать и модифицировать произвольные графические изображения. Кроме того, из соображений эффективности целесообразен ввод дополнительных, специальных типов Г-объектов, часто используемых на практике. К ним относятся:

— макеты таблиц;

— оси графиков и сами графики;

— столбчатые гистограммы;

— круговые (секторные) диаграммы;

— параллельные и перпендикулярные линии;

— сопрягающие дуги;

— геометрические размеры и размерные линии;

— текстовые надписи (строки алфавитно-цифровых символов);

— произвольные («проволочные») изображения, формируемые как след движения курсора по сетке экрана;

— позиционное меню, представляющее собой совокупность характерных точек на экране и предназначенное для выбора изображенных объектов.

С каждым из этих Г-объектов ассоциируются процедуры, обеспечивающие адекватные построения, например, ввод-вывод данных в таблицы (данные сохраняются в файлах отдельно от макетов таблиц); построение графиков, гистограмм и диаграмм по данным, извлекаемым из определенных файлов; правильные геометрические построения на основе понятий параллельности, перпендикулярности и сопряжения; нанесение-удаление размерных линий и масштабирование изображений на их основе; нанесение-удаление текстовых надписей; формирование и модификация позиционного меню.

Один из распространенных методов работы с видеоизображениями предполагает формирование библиотек Г-объектов (GL), в которых хранятся либо законченные изображения (экземпляры картин, таблиц, чертежей), либо условные графические символы, используемые при построении изображений логических схем, приборов, химических и других структур. В первом случае библиотечные Г-объекты выводятся на экран или на графопостроитель относительно заданного абсолютного базиса. Они служат, в основном, для быстрого воспроизведения заранее заготовленного, возможно, весьма сложного изображения. Во втором случае извлекаемые

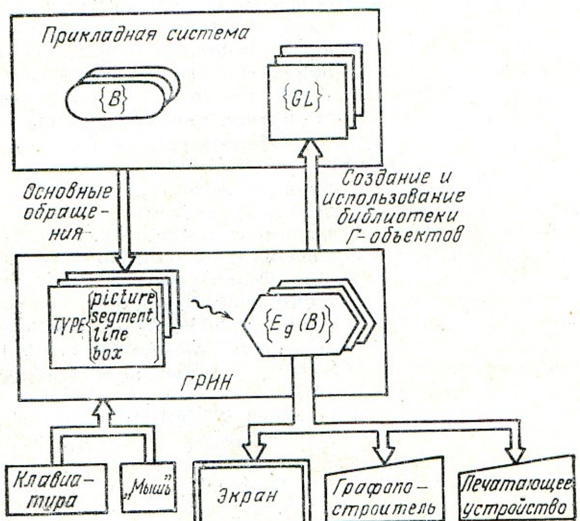


Рис. 2. Взаимодействие графического интерфейса (ГРИН) и прикладной системы

из GL условные графические символы, наоборот, служат для интерактивного построения изображений структурных объектов, сопровождаемого формированием семантической модели строящегося объекта.

Все процедуры графического интерфейса — как относящиеся к отдельным абстрактным типам, так и независимые — сгруппированы в несколько программных библиотек и могут вызываться из прикладной системы, оперирующей именами процедур и их параметрами или от пользователя через простую схему диалога, основанную на анализе кодов нажимаемых клавиш и (или) параметров графических манипуляторов «джойстик», «мышь», планшет. Схема взаимодействия графического интерфейса и прикладной системы представлена на рис. 2.

Адаптивная диалоговая среда. Взаимодействие человека и машины при решении прикладной задачи всегда протекает в рамках сценария, задуманного и реализованного создателем системы. В большинстве традиционных прикладных систем реализация такого сценария практически неотделима от реализации содержательной программной части, поскольку она осуществляется с помощью операторов ввода-вывода, размещенных в соответствующих точках программы.

Наш подход, в отличие от традиционного, предполагает, во-первых, *формальное описание диалогового сценария*, отделенное от содержательной части прикладной системы; во-вторых, *возможность его модификации*, что обеспечивает адаптивность системы, т. е. ее способность к динамической подстройке в соответствии с конкретными потребностями пользователей.

Основная информационная единица, описывающая этап диалога, называется *диалоговым состоянием* или Д-объектом. Множество Д-объектов {Д} определяет *схему диалога* или Д-схему. Общая Д-схема может быть разбита на подсхемы или *диалоговые процессы*, каждому из которых соответствует подмножество $\{D'\} \subseteq \{D\}$. Диалоговые состояния в рамках Д-схемы соединяются между собой переходами, образуя связный ориентированный граф.

В соответствии с общим подходом можно определить абстрактные типы D-schema и D-object, и ряд сопутствующих типов, на основе которых осуществляется программная реализация адаптивной диалоговой среды (АДС):

Д-схема имеет индивидуальное имя, доступное пользователю, ссылку на начальный Д-объект и ссылку на файл с пояснениями к данному диалоговому процессу. Каждый Д-объект также характеризуется внешним именем и типом, указывающим на способ его интерпретации. Обработка Д-объекта состоит из следующих шагов (некоторые шаги могут пропускаться, если нет соответствующего задания).

1. Выполнение предварительных действий — таких как очистка экрана, задание режима работы, вывод на экран некоторых начальных пояснений или подсказок и другие.
2. Инициация и открытие рабочего окна на экране дисплея.
3. Выполнение основных рабочих действий, таких как вывод в рабочее окно определенного текста или изображения, выдача звуковых сигналов, настройка на обработку меню и другие.
4. Обработка стандартной реакции пользователя, в которую входит перемещение курсора по рабочему окну с целью выбора объектов или позиций на экране, запрос дополнительных пояснений (подсказок), экстренное прерывание процесса и другие.
5. Обработка завершающей реакции пользователя, которая определяется типом данного Д-объекта.

Если Д-объект имеет тип «меню», то завершающая реакция состоит в нажатии пользователем одной из предопределенных функциональных или алфавитно-цифровых клавиш. Клавишам ставятся в соответствие варианты меню (см. тип menu-choice), каждый из которых включает действия по данному варианту и указание на следующее состояние данной схемы Д-схемы. Действия обычно обеспечивают обработку того объекта, на который в данный момент указывает курсор на экране, причем эта обработка может осуществляться стандартными системными функциями или прикладными программами (процедурами) пользователя. Действие может отсутствовать (быть пустым).

Для Д-объектов типа «запрос значения» предполагается, что пользователь в качестве завершающей реакции введет значение запрашиваемого системой параметра. Введенная величина подвергается первичному анализу с помощью простых предикатов (текст или число, больше, меньше, равно) и в результате происходит выбор одного из альтернативных вариантов (см. тип valalternative). Дальнейшая обработка альтерна-

```
TYPE D-schema=RECORD
      name: ALPHA;                          {имя Д-схемы}
      start: D-object;                      {начальное состояние}
      help: help-file;                      {файл с пояснениями (подсказкой)}
      END;
      D-obj-type-(menu, askval, form, dummy):
         {типы Д-объектов — меню, запрос значения, формат, холостой}
D-object=RECORD
      name: ALPHA;                          {имя Д-объекта}
      pred-actn: action;                    {предварительные действия}
      work-actn: action;                    {рабочие действия}
      work-window: window;                  {рабочее окно}
      type: D-obj-type;                     {тип Д-объекта}
      CASE D-obj-type OF;                   {варианты в зависимости от типа}
      menu: (m: ARRAY [. .] of menu-choice);
      askval: (v: ARRAY [. .] of val-alternative);
      form: (A: action; next: D-object);
      END;
menu-choice=RECORD                          {вариант меню}
      key: key-code;                        {код нажатой клавиши}
      act: action;                          {действия по данному варианту}
      nxt: D-object;                        {следующее состояние}
END;
val-alternative=RECORD                      {альтернативный вариант значения}
      pred: predicate;                      {предикат для оценки значения}
      act: action;                          {действия}
      nxt: D-object;                        {следующее состояние}
END;
```

тивного варианта значения аналогична обработке варианта меню.

Наконец, для Д-объектов типа «формат» предполагается, что пользователь после заполнения нескольких форматных полей нажимает клавишу, означающую завершение работы, после чего выполняются действия по обработке введенного формата. Как правило, такая обработка выполняется прикладной программой пользователя.

Обработка завершающей реакции обычно заканчивается переходом в следующее состояние Д-схемы, т. е. выбором следующего Д-объекта для обработки. Среди возможных действий в любых типах Д-объектов могут употребляться операторы вызова других, «вложенных» Д-схем, возврата из вложенных Д-схем на верхний уровень, полного завершения работы и другие. Кроме того, эти же действия могут инициироваться самим пользователем, для чего вместо обычных стандартных реакций ему достаточно нажать специально зарезервированные клавиши «спецреакций».

«Холостые» Д-объекты (типа dummy) используются для задания вспомогательных действий, без участия пользователя; при их обработке выполняются лишь шаги 1—3. Через состояния этого типа обеспечивается также запуск больших прикладных программ.

Объекты типа D-schema и D-object могут создаваться самим пользователем в ходе основной работы или перед ее началом, причем это осуществляется с помощью программы-конструктора Д-схем, основанной на использовании системы управления дисплейными окнами и графического интерфейса. Таким образом, реализуется принцип самоподдержки и адаптивности диалоговой системы.

Из приведенного описания должно быть ясно, что предложенный подход позволяет создавать Д-схемы, определяемые произвольными ориентированными графами и в то же время допускающими экстренные переходы, возвраты и прерывания предопределенных сценариев.

Объектная база. Рассмотренные выше компоненты системы основываются на использовании нескольких базовых понятий, представляемых в виде *абстрактных типов* — описаний структурных объектов и ассоциированных с ними процедур. При реализации абстрактные типы воплощаются в системные программы и данные и, следовательно, никак «не видны» пользователям. Внешний же облик прикладной системы определяется совокупностью конкретных структур, являющихся экземплярами абстрактных типов. Экземпляры становятся «осязаемыми» объектами, если их поместить в соответствующее хранилище и иметь возможность вызова, модификации и активизации.

Для хранения конкретных экземпляров системных объектов используется объектная база — специальная совокупность системных программ, обеспечивающая хранение и доступ к структурным данным. Природа хранимых объектов безразлична, и доступ к ним одинаков как со стороны системных, так и прикладных программ.

Кроме хранения системных объектов, отображающих дисплейные окна, Г-объекты и Д-объекты, в объектной базе хранятся также проблемно-ориентированные объекты {B}, несущие семантическую нагрузку. Для разных предметных областей наборы этих объектов различны, но может быть предложена общая таксономия, имеющая универсальный характер [1]. Могут быть выделены следующие типы проблемно-ориентированных объектов (ПОБ):

описатели классов (устройство, микросхема, плата);

описатели свойств (размеры, потребляемая мощность, число контактов);

описатели отношений (соединение, расстояние, передача сигнала);

конкретные экземпляры (микросхема *M*11, плата П3);

фактические отношения между ПОБ (сигнал *DS*, соединение *L*12);

описатели универсальных (терминальных) величин (тактовая частота, имя файла).

Для внутреннего представления этих типов удобно использовать объекты со списками свойств. Общий вид такого объекта в соответствии с данными работы [1]:

n: p1 v1; p2 v2 . . . ; pk vk,

где *n* — имя объекта; пары (pi vi) — свойства, причем pi — наименование i-го свойства, vi — его значение. Такой объект может быть представлен, в соответствии с общим подходом, абстрактным типом с ассоциированными процедурами (функциями):

```
TYPE property=RECORD                                        {свойство общего типа}
     p: ALPHA;                                              {имя свойства}
     v. prop--value;                                        {значение свойства}
     END;
P-object=RECORD                                             {пробл.-ориент. объект}
     n: ALPHA;                                              {имя объекта}
     t: obj-type;                                           {тип объекта}
     k: INTEGER;                                            {число свойств}
     pp: ARRAY [1 . . k] of property;                       {список свойств}
     END;
FUNCTION Put-prop (VAR n: P-object; p: ALPHA; v: prop-value): BOOLEAN; {занести свойство в объект}
FUNCTION Get-prop (n: P-object; p: ALPHA): prop-value;                 {извлечь значение свойства}
FUNCTION Del-prop (VAR n:P-object; p: ALPHA): BOOLEAN;                 {удалить свойство из объекта}
FUNCTION Find-obj (p: ALPHA; v-prop-value): ARRAY [. .] OF P-object;  {поиск объектов по заданным
                                                                        свойствам}
```

Необходимы также другие процедуры для полной поддержки работы с ПОБ.

Специальный механизм, описанный в работах [1] и [6], обеспечивает «наследование свойств» от объектов, являющихся классами, к объектам, являющимся подклассами или экземплярами классов. Этот механизм основан на использовании специального свойства, играющего роль указателя от вида к роду (или от подкласса к классу, от экземпляра к классу). При этом в видовом понятии или экземпляре считаются автоматически заданными все свойства, указанные хотя бы в одном из вышестоящих родовых понятий. Например, объект, описывающий конкретную микросхему, автоматически наследует все свойства (мощность, число контактов, логическую функцию и т. д.), заданные в описателе данного класса микросхем.

Еще один важный механизм связан с использованием законов {R}, ассоциируемых с ПОБ. В работах [1] и [6] описана общая структура объекта, интерпретируемого системой как закономерность (правило), справедливое для заданной предметной области:

pattern : condition => action

Здесь pattern — образец (шаблон), позволяющий сопоставлять обрабатываемое выражение с текущим состоянием модели предметной области; condition — условие, удовлетворение которого позволяет применить данное правило; action — последовательность операций,

выражающая требуемые действия над ПОБ. Среди операций одного правила могут употребляться обращения к другим правилам, что обеспечивает рекурсивность обработки. Наконец, самое важное это то, что имеются операции вызова любых прикладных программ, оформленных соответствующим унифицированным образом.

Применение

Рассмотренный метод организации человеко-машинного взаимодействия на основе интегрированных универсальных компонентов дает возможность прикладным программистам быстро создавать, а пользователям — легко осваивать конкретные прикладные системы. Так, например, разработчик электронных модулей (плат) пользуется графическим интерфейсом сначала для построения эскиза логической схемы модуля, а затем — для физического размещения корпусов микросхем на плате и проведения соединяющих линий. Система управления дисплейными окнами вместе с текстовым процессором позволяет задать для каждого компонента необходимые характеристики, а адаптивная диалоговая среда — общую и частные схемы взаимодействия с системой на всех этапах проектирования. При этом отдельные задачи могут решаться унифицированными проблемно-ориентированными программами, например, расчет мощности, частичная трассировка плат, генерация тестов и логическое тестирование. В объектной базе при этом хранятся как системные объекты, поддерживающие все процессы, так и проблемно-ориентированные, подобные тем, что были упомянуты в примерах.

Оператор при общении с такой системой постоянно работает в одной и той же операционной среде. Это проявляется, например, в том, что для перемещения курсора на экране, выбора, удаления, перемещения объектов используются одни и те же клавиши, независимо от природы и вида объектов. Весь процесс протекает в рамках диалогового сценария, который имеет внешнее описание и легко изменяется.

ЛИТЕРАТУРА

1. Брябрин В. М. Ф-язык — формализм для представления знаний в интеллектуальной диалоговой системе. — В кн.: Прикладная информатика. — М.: Финансы и статистика, 1981, вып. 1, с. 73—103.

2. Кахро М. И., Калья А. П., Тыугу Э. Х. Инструментальная система программирования ЕС ЭВМ (ПРИЗ). — М.: Финансы и статистика, 1981. — 156 с.

3. Бабаев И. О., Новиков Ф. А., Петрушина Т. И. Язык ДЕКАРТ — входной язык системы СПОРА. — В кн.: Прикладная информатика. М.: Финансы и статистика, 1981, вып. 1, с. 35—72.

4. Родин С. Р., Швалев А. П., Эрлих А. Т. МАВР — диалоговая система моделирования альтернатив и выбора решений. — Труды междун. симпозиума «Искусственный интеллект и информационно-управляющие системы роботов», Братислава, 1982, с. 71—73.

5. Брябрин В. М. Адаптивный диалог — основа персональной вычислительной системы. — В кн.: Лингвистические процессоры и представление знаний. — Новосибирск: ВЦ СО АН СССР, 1981, с. 22—40.

6. Пономарев В. В. Диалоговая система, базирующаяся на методе К-единиц. Обработка символьной информации. — М.: ВЦ АН СССР, 1984, с. 60—71.

Статья поступила 23 апреля 1984 г.

УДК 681.3—181.48

М. Г. Кыпп, Э. Х. Тыугу

ОБЪЕКТНО-ОРИЕНТИРОВАННАЯ СИСТЕМА ПРОГРАММИРОВАНИЯ ДЛЯ МАЛЫХ ЭВМ

Система программирования МИКРОПРИЗ, входной язык которой близок к языку УТОПИСТ [1], работает на микроЭВМ Apple II с 8-битным микропроцессором и с 64 Кбайтами памяти. Она написана полностью на языке Паскаль и поэтому легко переносима на другие малые ЭВМ.

Проект, по которому разрабатывалась система МИКРОПРИЗ, имел первоначально следующие цели:

проверка практической целесообразности применения структурного синтеза программ на микроЭВМ;

получение данных о возможности реализации объектно-ориентированной системы программирования на языке Паскаль.

Выяснилось, что благодаря упрощенному входному языку и более экономному внутреннему языку, МИКРОПРИЗ практически не уступает системе ПРИЗ [2] по скорости трансляции и синтеза программ. Поэтому было принято решение довести систему программирования МИКРОПРИЗ до такой степени готовности, чтобы ей могли пользоваться непрофессиональные программисты. Ориентация на пользователя, не являющегося программистом, является важной чертой системы МИКРОПРИЗ.

Рассматриваемая в данной статье версия системы отличается от практически эксплуатируемой только лексикой (приложение 1), которая заменена в статье на русскую.

Модель системы для пользователя

С точки зрения пользователя МИКРОПРИЗ является решателем задач, обладающим *базой знаний*, в которой хранятся описания *понятий*. Для работы с базой знаний предусмотрены команды !ОПРЕДЕЛИТЬ !КАТАЛОГ !ЗНАЧЕНИЕ !ИСКЛЮЧИТЬ !БАЗА.

По команде !ОПРЕДЕЛИТЬ производится обучение системы новым понятиям. Остальные команды имеют очевидный смысл: выдача каталога, выдача описания понятия, исключение понятия и создание новой базы знаний. Поскольку при работе с базой знаний диалогом управляет система, пользователю достаточно помнить только перечисленные выше команды — если в ходе выполнения команды требуются его вмешательство, система сообщает об этом.

Команда !СПРАВКА позволяет получить от системы инструкцию для работы, в частности, список всех команд с пояснениями. По команде !ФАЙЛ входной текст берется из файла, имя которого задается пользователем. Это позволяет использовать для подготовки программ стандартны средства редактирование текста и хранить входные тексты в файлах. Для обучения пользователя имеются текстовые файлы с комментариями, при подаче которых на вход системы можно проследить все возможности работы.

По команде !РЕШЕНИЕ система переходит в режим решения задач. По составленной пользователем спецификации — модели задачи. При этом могут использоваться понятия, имеющиеся в базе знаний.

Язык описания задач и язык представления знаний для обучения системы — два различных языка. В принципе, каждый пользователь может создать свой собственный язык, записывая в базу знаний соответствующие понятия. Создание нового языка может оказаться сложным, главным образом, из-за трудностей определения новых понятий через совокупность функций, которые должны быть описаны. Однако, существуют классы задач, для которых специализированные языки создаются достаточно просто, например, задачи по геометрии, элементарной физике, электротехнике, расчеты прочности, задачи моделирования динамических систем. Примеры таких задач приведены ниже.

Описание задач

Пользователь задает модель задачи, являющуюся для системы МИКРОПРИЗ спецификацией программы, которую необходимо автоматически построить. Спецификация представляет собой последовательность описаний и уравнений на входном языке. На внутреннем языке она принимает форму совокупности объектов и аксиом. Описание полного языка спецификации приведено в приложении 2.

С точки зрения пользователя модель задачи является совокупностью переменных и функциональных зависимостей, заданных уравнениями или программами. В простейшем случае — это совокупность уравнений, заданных прямо в описании задачи. Например, имея текст:

X+Y=2
Y×Z=1
X=1,

можно по команде ВЫЧИСЛИТЬ решить систему уравнений, и получить ответ:

→ X=1
→ Y=1
→ Z=1.

Уравнения решаются численным методом, системы уравнений — последовательной подстановкой переменных.

Значительная экономия труда пользователя достигается при использовании в модели ранее описанных понятий, например элементарных геометрических фигур (рис. 1). Тогда совсем просто описываются задачи на вычисление отдельных элементов фигур, площадей и т. п., не требующие дополнительных построений:

К :КВАДРАТ A=1
С :КРУГ D=K.A
S=K.S—C.S

К и С — имена конкретных объектов: квадрата и круга, вводимых в модель задачи. По описанию объекта в модель вводится переменная, обозначающая объект, а также все переменные, имеющиеся в описании используемого понятия в базе знаний. Для квадрата это переменные A, D, S, P. Их имена дополняются именем объекта в виде префикса. Таким образом, после перевода первой строки спецификации в модели задачи будут переменным K.A, K.D, K.S, K.P и переменная K, обозначающая квадрат целиком. Кроме того, в модель задачи будут введены отношения

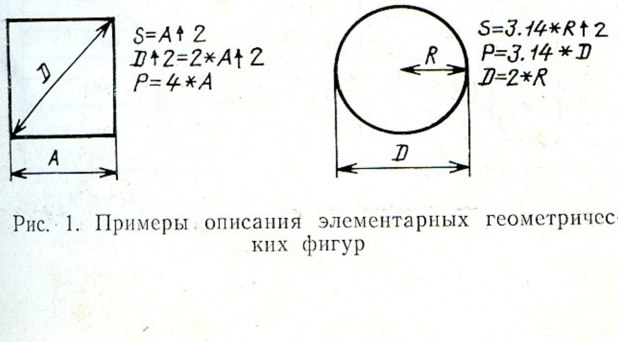


Рис. 1. Примеры описания элементарных геометрических фигур

K.S=K.A ∧ 2
K.D ∧ 2=2×K.A ∧ 2
K.P=4×K.A
K.A.=1.

Первые три уравнения получаются из описания понятия квадрат. Аналогично переводится вторая строка. Она дает перменные: C.D, C.R, C.S, C.P, C и уравнения:

C.S=3.14×C.R>2
C.P=3.14×C.D
CD=2×C.R
C.D=K.A.

Добавив к ним уравнение из спецификации задачи, получим систему уравнений, из которой легко вычислить требуемую площадь S.

В описании задачи можно определять структуры. Например,

ВОЗРАСТ:ЧИСЛО
ФАМИЛИЯ:ТЕКСТ
АДРЕС:ТЕКСТ
ОТДЕЛ:ТЕКСТ
СОТРУДНИК: (ФАМИЛИЯ, АДРЕС, ВОЗРАСТ, ОТДЕЛ)

является описанием структуры СОТРУДНИК.

При описании понятий могут быть использованы программы, а у отношений, выраженных программами, допускаются подзадачи. Например, по аналогии с системой ПРИЗ [2], могут быть описаны понятия множества и подмножества. В частности, задачи распечатки сотрудников с возрастом более 30 лет из файла 'СОТР' задается спецификацией:

R: ПОДМНОЖЕСТВО ЭЛЕМЕНТ=СОТРУДНИК, ФАЙЛ='СОТР'
R. УСЛОВИЕ=ВОЗРАСТ—30

Понятие ПОДМНОЖЕСТВО позволяет использовать простые базы данных и является существенным расширением входного языка. Кроме того, в язык описания задач включены некоторые встроенные понятия, например ТАБЛИЦА и МАКСИМУМ. Первое необходимо для циклического выполнения вычислений и выдачи результатов в табличном виде, второе — для вычисления максимума функций на заданном интервале. Оба понятия определены через программно-реализованные отношения с подзадачами.

Режим вычислений задается одной из следующих команд:

ВЫЧИСЛИТЬ
ВЫЧИСЛИТЬ X, . . . , Y
ВЫЧИСЛИТЬ X, . . . , Y ПО U, . . . , V,

где X, . . . , Y, U, . . . , V — переменные.

По первой команде вычисляются и выводятся на экран значения всех переменных, которые могут быть вычислены. По второй — только значения X, . . . , Y и тех переменных, которые необходимы для этого (выводятся на экран только значения для X, . . . , Y). По третьей команде строится программа вычисления X, . . . , Y по U, . . . , V, которая запрашивает значения U, . . ., V и вычисляет X, . . ., Y. Если при использовании второй и третьей команд некоторые переменные X, . . , V не могут быть вычислены, то об этом выдается сообщение.

После выполнения вычислений можно либо продолжить описание модели задачи, сохранив в ней результаты вычислений, либо стереть модель командой !НОВАЯ МОДЕЛЬ, и начать описание новой задачи. Использование трех режимов вычислений на постепенно наращиваемой модели позволяет в диалоге достаточно гибко решать многие вычислительные задачи.

Язык представления знаний

Новые понятия описываются на языке представления знаний. При описании понятия задается его имя:

НОВЫЙ <идентификатор>,

затем предложениями

ПАРАМ <индентификатор> : <тип>

описываются все переменные и структуры, входящие в его описание. После этого задаются уравнения, и, наконец, описываются отношения, представляемые программами.

Например, понятие КВАДРАТ может быть описано текстом:

```
НОВЫЙ КВАДРАТ
ПАРАМ A:ЧИСЛО
      D:ЧИСЛО
      S:ЧИСЛО
      P:ЧИСЛО
УРАВ  A∧2=S
      D∧2=2×A∧2
      P=4×A.
```

Параметры могут иметь типы: ЧИСЛО, ТЕКСТ, НЕОПРЕДЕЛЕННЫЙ, а также могут быть структурами, аналогичными структурам языка описания задач. Типы ЧИСЛО и ТЕКСТ такие же, как в языке описания задач. НЕОПРЕДЕЛЕННЫЙ является универсальным типом. Параметр такого типа может связываться с любой переменной, в том числе, со структурой или с объектом (приложение 2). Это позволяет использовать понятия, параметризованные типами их переменных. В качестве примера параметризованного понятия рассмотрим описание предопределенного понятия ТАБЛИЦА.

```
НОВАЯ ТАБЛИЦА
ПАРАМ АРГУМЕНТ : ЧИСЛО
      СТРОКА : НЕОПРЕДЕЛЕННАЯ
      ОТ : ЧИСЛО
      ДО : ЧИСЛО
      ШАГ : ЧИСЛО
      РЕЗУЛЬТАТ : ТЕКСТ
МОДУЛЬ ТАБЛ
ПОДЗАДАЧА АРГУМЕНТ→СТРОКА
ВХ ОТ
ВХ ДО
ВХ ШАГ
ВЫХ РЕЗУЛЬТАТ.
```

Последние шесть строк этого описания представляют отношение, заданное программой ТАБЛ, насчитывающей четыре входных параметра: процедуру вычисления значения структуры СТРОКА по значению АРГУМЕНТ, и числовые параметры: ОТ, ДО, ШАГ. Выходным параметром является текстовая переменная РЕЗУЛЬТАТ. Значение первого входного параметра (процедура вычисления структуры СТРОКА) строится автоматически в ходе планирования вычислений. Это — пример подзадачи, решение которой планируется каждый раз, когда это необходимо для выполнения команды ВЫЧИСЛИТЬ или какой-нибудь другой команды, задающей режим вычислений.

При использовании понятия ТАБЛИЦА вместо параметра СТРОКА представляется конкретная структура, значения которой печатаются в виде строк таблицы. Например, спецификация

```
X : ЧИСЛО
Y : ЧИСЛО
C : (X, Y)
Y=X∧2
T : ТАБЛИЦА ОТ=0, ДО=10, ШАГ=1, СТРО-
    КА=C, АРГУМЕНТ=X
```

описывает таблицу квадратов целых чисел от 0 до 10.

Другим примером параметризованного понятия является ПОДМНОЖЕСТВО:

```
НОВОЕ ПОДМНОЖЕСТВО
ПАРАМ ЭЛЕМЕНТ : НЕОПРЕДЕЛЕННЫЙ
      ФАЙЛ : ТЕКСТ
      РЕЗУЛЬТАТ : ТЕКСТ
      УСЛОВИЕ : ЧИСЛО
МОДУЛЬ ПОДМН
ПОДЗАДАЧА ЭЛЕМЕНТ → УСЛОВИЕ
ВХ ФАЙЛ
ВЫХ РЕЗУЛЬТАТ
```

Модуль ПОДМН, получив на входе имя файла, указывающее откуда производить выборку, и процедуру вычисления значения УСЛОВИЕ, печатает значения переменной ЭЛЕМЕНТ, для которых УСЛОВИЕ>0 (поскольку в языке нет логических переменных, логические условия должны моделироваться арифметическими выражениями).

Внутренний язык

Заданная пользователем спецификация программы (описание задачи) на внутреннем языке принимает форму совокупности объектов и аксиом. В целях экономии аксиомы группируются в отношении. Например, аксиомы уравнения $f(x_1, \ldots, x_m) = g(y_1, \ldots, y_n)$ представляются в виде *симметричного отношения* [3], связывающего переменные $x_1, \ldots, x_m$, $y_1, \ldots, y_n$. Аксиомы структуры x: $(x_1, \ldots, x_m)$ представляются в виде *структурного отношения* [3] между переменными x, $x_1, \ldots, x_m$.

Внутреннее представление модели задачи должно обеспечить однопроходную трансляцию и быстрое планирование линейных участков. Поэтому *модель представляется в виде таблиц объектов, отношений и связей* (рис. 2). Таблицы строятся по ходу трансляции описания задачи. Размеры модели задачи ограничены размерами таблиц.

Таблица объектов содержит для каждого объекта следующую информацию: идентификатор, тип, значение или ссылку на значение, признак наличия значения, ссылку на таблицу связей.

Таблица отношений включает по одной строке для каждого отношения (уравнения, функционального отношения, подзадачи или структурного отношения). В ней хранятся идентификатор отношения, тип, ссылка на реализацию, счетчик входных переменных и ссылка на таблицу связей.

Таблица связей хранит информацию о связях отношений с объектами, в том числе тип связи. Непосредственно в начале планировки строится и четвертая таблица, содержащая информацию о связях объектов с отношениями, полезная для ускорения планировки.

При планировании решения задачи используются правила структурного синтеза программ с независимыми подзадачами [4]. Результатом планирования являет-

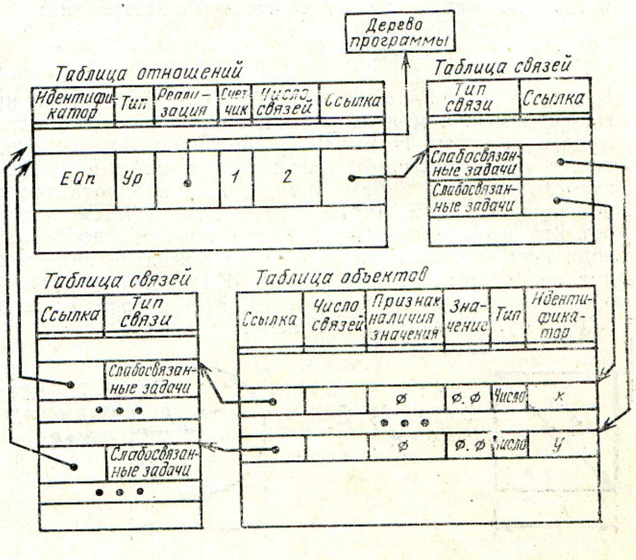


Рис. 2. Представление модели задачи пользователя в виде таблиц

ся алгоритм решения задачи в символьном виде, имеющий структуру дерева. Алгоритм выполняется интерпретатором, который является программой прохождения дерева. В каждой вершине, соответствующей какому-нибудь отношению, интерпретатор вызывает нужные программы для выполнения отношения.

Базы знаний находятся на диске и организованы в виде таблиц подобно внутреннему представлению модели задачи. По описанию X : T в таблицы модели копируются строки, соответствующие параметрам и отношениям понятия T, из таблиц используемой базы знаний, и вносятся нужные изменения.

Приложение 1

Соответствие русской и английской лексики

Директивы

!ОПРЕДЕЛИТЬ	— !DEFINE
!КАТАЛОГ	— !CATALOG
!ЗНАЧЕНИЕ	— !MEANING
!ИСКЛЮЧИТЬ	— !X
!БАЗА	— !ARCHIVE
!СПРАВКА	— !HELP
!ФАЙЛ	— !FILE
!РЕШЕНИЕ	— !SOLVE
!НОВАЯ МОДЕЛЬ	— !NEW

Зарезервированные идентификаторы

ВЫЧИСЛИТЬ	— COMPUTE
ПО	— FROM
ТАБЛИЦА	— TABLE
МАКСИМУМ	— MAX
НОВЫЙ	— NEW
ПАРАМ	— PAR
ЧИСЛО	— NUMERIC
ТЕКСТ	— TEXT
НЕОПРЕДЕЛЕННАЯ	— UNDEFINED
УРАВ	— EQN
МОДУЛЬ	— MOD
ПОДЗАДАЧА	— SUB
ВХ	— IN
ВЫХ	— OUT

Директивы распознаются по первой букве, зарезервированные идентификаторы по первым трем буквам.

Приложение 2

Язык спецификаций

Спецификация является последовательностью описаний и уравнений. На внутреннем языке она представляет собой множество объектов и множество аксиом в виде предложений вычислимости [4]. Семантику описаний и уравнений можно задать в виде совокупности тех объектов и аксиом, которые порождаются при их переводе на внутренний язык.

Описание <ид> : <описатель> задает объект, получающий имя, заданное идентификатором слева от «:». Описатель определяет свойства объекта.

<описатель> : : = { <первичный> | <структура> | <имя понятия> [<переделки>] }

<первичный> : : = { ЧИСЛО | ТЕКСТ }.

Первичными объектами являются числовые (целые и десятичные) и текстовые переменные.

Структура составляется из компонентов, которыми являются ранее описанные объекты. Их имена перечисляются в скобках:

<структура> : : = (<имя> [,<имя>] ...).

По описанию структуры x: $(x_1, \ldots, x_k)$ создается новый объект x, и задаются аксиомы:

$$x \to x_1; \; \ldots \; x \to x_k; \; x_1, \ldots, x_k \to x.$$

Описание, содержащее имя понятия, задает новый объект. По описанию X : T, где T — имя понятия, создается объект с именем X и для него задаются все аксиомы, имеющиеся в описании понятия T. Компоненты объекта, имеющие в описании T имена A, B..., получают имена X.A, X.B, ...

Переделки при использовании имени понятия задаются равенствами x=y или x=<значение>, где x — имя компонента (параметра) используемого понятия, y — имя ранее определенного объекта. Переделка x=y задает две аксиомы: $x \to y$ и $y \to y$, а переделка x=<значение> задает одну аксиому: $\to x$.

Уравнения имеют две формы:

$$E(x_1, \ldots, x_k) = F(x_1, \ldots, x_k);$$

x=<значение> или x=y.

В первом случае E и F — арифметические выражения, содержащие имена $x, \ldots, x_k$ объектов числового типа. Во втором — x, y — имена объектов нечислового типа. По уравнению создаются аксиомы

$$x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_k \to x_i \text{ для } i = 1, \ldots, k.$$

Семантикой (семантическим значением спецификация является объединение множеств объектов, порожденных описаниями, и объединение множеств аксиом, порожденных описаниями и уравнениями.

$X_1 : T_1 \pi_1;$

.
.
.

$X_k : T_k \pi_k;$

$Y_1 : (A_1, \ldots, B_1);$

.
.
.

$Y_m : (A_m, \ldots, B_m);$

$E_1 = F_1;$

.
.
.

$E_n = F_n;$

где T_i — первичный описатель или имя класса, π_i — пусто или переделки, $A_j, \ldots, B_j$ — имена объектов, E_s, F_s — арифметические выражения, X_i, Y_j — различные идентификаторы, $i = 1, \ldots, k$, $j = 1, \ldots, m$, $s = 1, \ldots, n$. Полученное таким образом, множество аксиом, вместе с правилами вывода структурного синтеза программ с независимыми подзадачами, образует теорию τ. Теоремы этой теории $y_1, \ldots, y_m$ соответствуют разрешимым задачам

ВЫЧИСЛИТЬ $y_1, \ldots, y_m$.

ЛИТЕРАТУРА

1. Мяннисалу М. А., Тыугу Э. Х., Унт М. И., Фуксман А. Л. Язык УТОПИСТ. — Алгоритмы и организация решения экономических задач, 1977, вып. 10, с. 80—118.

2. Кальял А. П., Кахро М. И., Тыугу Э. Х. Инструментальная система программирования ЕС ЭВМ (ПРИЗ). — М.: Финансы и статистика, 1981. — 158 с.

3. Тыугу Э. Х. Концептуальное программирование. М.: Наука, 1984.

4. Тыугу Э. Х., Харф М. Я. Алгоритмы структурного синтеза программ. — Программирование, 1980, № 4, с. 3—13.

Статья поступила 22 марта 1984 г.

УДК 61:658

Г. С. Цейтин

ПСИХОЛОГИЧЕСКИЕ ЗАДАЧИ В РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Программное обеспечение всегда разрабатывается в расчете на использование его человеком и должно учитывать требования удобства и эффективности его использования. Эти вопросы должны решаться наравне с чисто техническими вопросами, обусловленными конкретным характером решаемых задач и их отображением на ресурсы реального оборудования; в широком классе задач, относимых обычно к области системного программного обеспечения, организация взаимодействия с пользователем вообще выдвигается на первое место. Трудности обычно состоят не в том, чтобы запрограммировать желаемое поведение машины в таком взаимодействии, а в том, чтобы решить, какое именно поведение для нас желательно. Таким образом, любая разработка программного обеспечения включает в себя задачу проектирования деятельности будущего пользователя создаваемой системы, а в задачах системного программирования такому проектированию принадлежит определяющая роль.

Возникающие здесь вопросы родственны традиционным задачам инженерной психологии, решаемым, например, при разработке пультов управления, но, естественно, сложнее, как ввиду более широких возможностей универсальных ЭВМ, так и потому, что эти задачи теснее связаны с мышлением пользователя, структурой и функционированием его знаний, наконец, требуют учета социальных факторов (например, при взаимодействии многих пользователей в рамках одной системы).

Разработчики программного обеспечения встречаются с этими вопросами в своей повседневной практике, но решают их в лучшем случае на основе накопившегося опыта (порой, правда, немалого). Иногда же эти вопросы решаются на основе случайных соображений или вообще не осознаются должным образом. Недостатки, выявившиеся впоследствии при реальной эксплуатации системы, исправлять труднее, а кроме того, пользователи часто тоже недостаточно квалифицированы для того, чтобы осознать и правильно сформулировать свои трудности, не говоря уже о рекомендациях по исправлению системы.

Необходимость теоретической разработки этих вопросов с позиций психологии уже давно назрела (автор впервые пытался привлечь к этому внимание психологов более 15 лет назад). Однако, насколько можно судить, психология не встречалась прежде с задачами этого класса и сегодня не только не может дать конкретные практические рекомендации, но даже не предлагает общей теоретической основы, в рамках которой можно было бы выделять конкретные исследовательские задачи и ставить эксперименты. Цель настоящей статьи — привлечь внимание к описанной проблематике, перечислив некоторые из часто встречающихся проблем. Наряду с вопросами, специфичными для разработки программного обеспечения, здесь упомянуты и вопросы более общего характера, которые, вероятно, уже исследовались психологами. Это перечисление не претендует на роль исчерпывающей научной классификации, а предлагаемая группировка проблем весьма условна. Разработчики программного обеспечения легко узнают в этом списке многие из своих собственных проблем.

Далее под системой понимается тот комплекс программ, данных и оборудования вместе с правилами их использования, в связи с разработкой которого ставятся рассматриваемые проблемы. Пользователем в данном контексте считается любое лицо, выполняющее действия, предписанные правилами использования этой системы, независимо от того, самостоятельно ли оно ставит задачу, требующую обращения к этой системе. Если в перечислении проблем просто упоминается некоторая возможность, то имеется в виду, что она осуществима технически, но нужно выяснить условия, в которых целесообразно ее использование.

1. Общие характеристики системы по отношению к пользователю

1.1. Ориентация системы на тот или иной уровень подготовки пользователя, на регулярного или случайного пользователя, на наличие или отсутствие у него прежнего опыта работы с этой системой. Возможность учета индивидуальных различий между пользователями в этих отношениях.

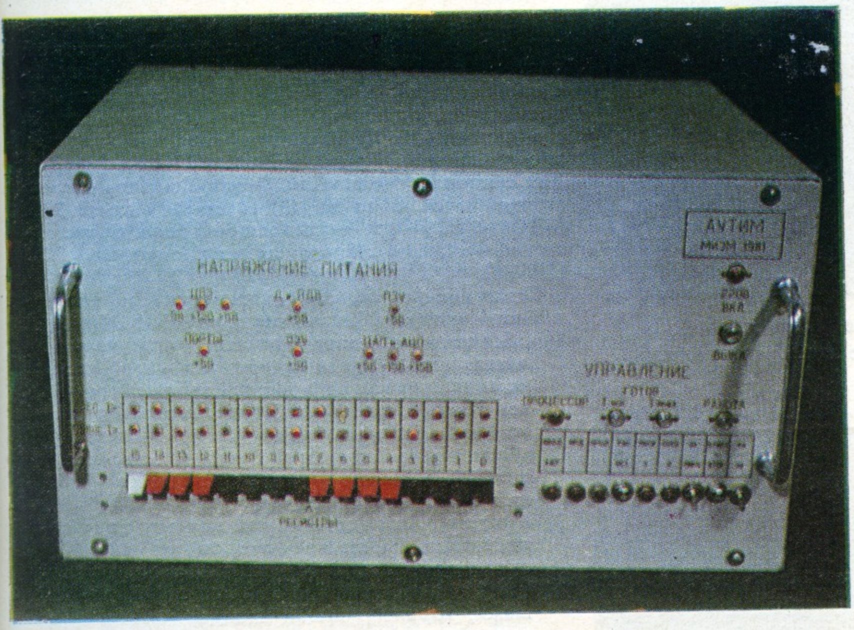
Распространенная ошибка в решении этих вопросов — приписывание будущему пользователю системы собственных качеств. Например, руководитель, который будет принимать систему, но фактически не будет работать на ней сам и даже некомпетентен для этого, часто требует дробления работы системы на мелкие шаги с выдачей из системы избыточно подробных разъяснений на каждом шаге при одновременном сведении действий, требуемых от пользователя, к самым простым. Системный программист, напротив, часто склонен не только сокращать до предела выдаваемую системой информацию и перекладывать значительную часть инициативы на пользователя (что, как показал опыт, в конечном счете приемлемо после некоторой тренировки), но и оставлять неразъясненными некоторые неполадки, не связанные собственно с работой системы, например, временную неготовность какого-либо элемента оборудования (усилия по преодолению таких неполадок порой незаметны для опытного системного программиста, а простой пользователь может на этом надолго застрять, не понимая, что происходит).

1.2. Учет возможностей неправильных действий пользователя. Выбор точки равновесия между максимальной свободой действий для пользователя (при которой он может нанести существенный ущерб системе и самому себе) и созданием предельно узких рамок, ограничивающих его действия (что может существенно затруднить использование системы, особенно в нестандартных случаях). Учет индивидуального уровня самоконтроля пользователя. Выделение ситуаций, требующих дополнительного контроля со стороны системы. Форма предъявления пользователю сообщений об ошибке и степень подробности разъяснения этих ошибок. Построение системой гипотез о причине и локализация ошибки пользователя (реально система обнаруживает не ошибку, а несоответствие между несколькими объектами, и заранее не ясно, какой именно из них ошибочен). Допустимы ли попытки системы самостоятельно исправить ошибки пользователя (с подтверждением со стороны пользователя или с последующим уведомлением)? Целесообразно ли уточнять локализацию ошибки в режиме диалога? Классификация ошибок по степени серьезности. Надо ли отделять информацию об ошибках от остальной выдачи или помещать ее вместе с обычной информацией, выдаваемой в том же месте, где обнаружена ошибка (например, включать информацию об ошибках в программе внутрь распечатки ее текста)?

1.3. Свобода выбора для пользователя. Предъявление пользователю инструкций в форме заданной последовательности действий на каждый случай или в виде перечисления в

СИСТЕМА УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМИ ПРОЦЕССАМИ НА ОСНОВЕ КОМПЛЕКТА БИС КР580

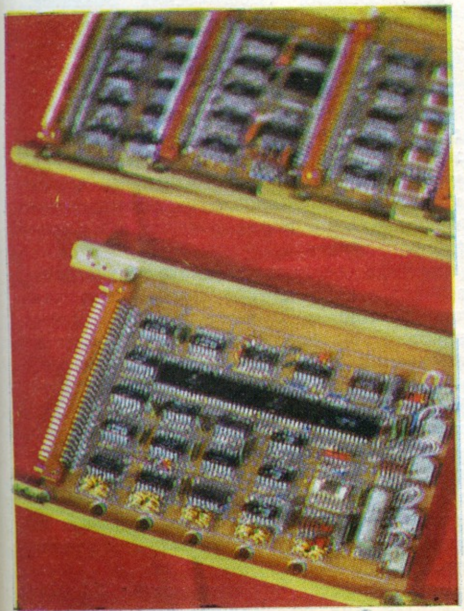
(к ст. Н. П. Давыдова, Ж. К. Давыдовой, Г. Т. Мишина и др.)



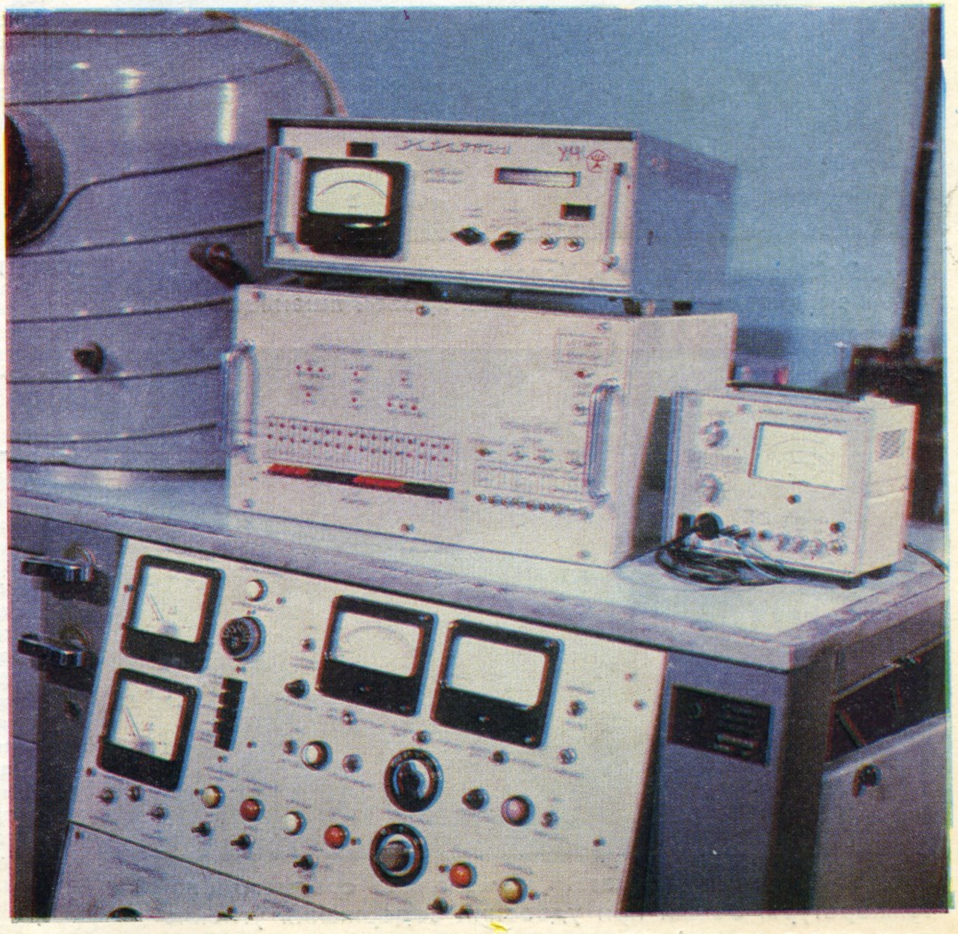
Общий вид контроллера АУТИМ

Основа **контроллера АУТИМ** — цифровая микроЭВМ на базе микропроцессора КР580ИК80А. АУТИМ управляет технологическими процессами микроэлектроники, например, термическим испарением и конденсацией в вакууме при изготовлении тонкопленочных прецизионных резистивных элементов и микросборок на их основе

Модули АУТИМа. На переднем плане модуль программно-управляемого цифрового измерителя сопротивлений



АУТИМ в составе вакуумной напылительной установки



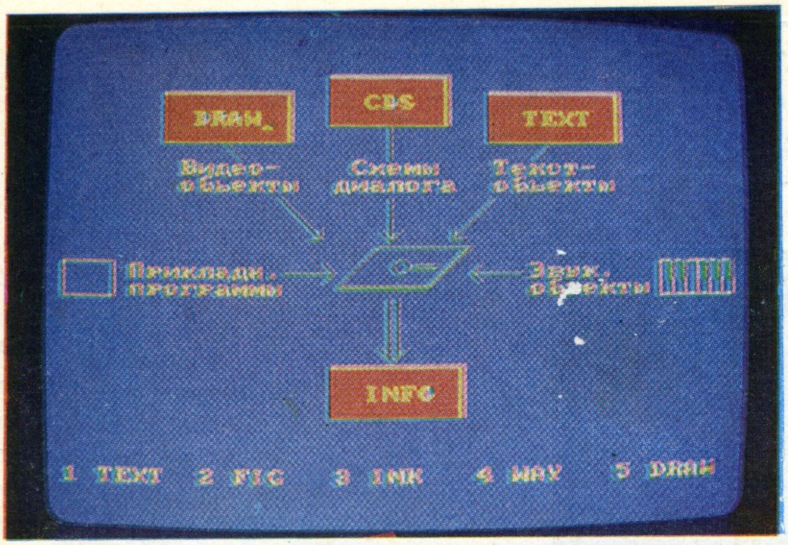


Рис. 1. Структурная схема конструктора диалоговых систем, позволяющего проектировать проблемно-ориентированные схемы диалога



Рис. 2. Система «Зерно». Условная карта СССР с отображением динамики уборки урожая яровой пшеницы

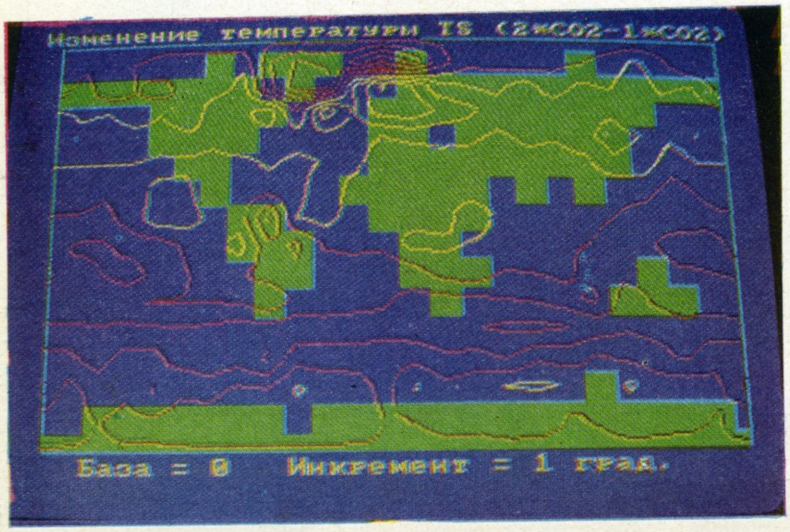


Рис. 3. Система «Климат». Изменение среднегодовой температуры Земли при удвоении содержания углекислого газа в атмосфере

В. М. Брябрин

ИНТЕЛЛЕКТУАЛЬНЫЙ ЧЕЛОВЕКО-МАШИННЫЙ ИНТЕРФЕЙС

В Вычислительном Центре АН СССР разработан метод построения прикладных диалоговых систем, основанный на использовании интеллектуального интерфейса—особой совокупности программ, позволяющей отделять пользователя от стандартного программного обеспечения, давая ему удобный и разнообразный набор средств для общения с машиной. В течение ряда лет этот метод опробован на больших и малых ЭВМ (БЭСМ-6, СМ-4) и в настоящее время реализуется на профессиональных персональных ЭВМ (ПЭВМ).

Основные принципы организации интеллектуального интерфейса на персональных ЭВМ:

1) обеспечение разнообразных форм человеко-машинного взаимодействия — на основе командных языков, меню, таблиц, графиков, текстовых документов, произвольных графических изображений;
2) обеспечение единства стиля и максимальной простоты действий пользователя при общении с системой;
3) интеграция представления данных, предусматривающая единство методов их внутренней организации при разных видах обработки;
4) унификация связей с прикладными программами, гарантирующая стандартизованный метод стыковки универсальных и проблемно-ориентированных программных модулей.

В число программных компонентов интеллектуального интерфейса входят в настоящее время:

СУОК — система управления дисплейными окнами, которая обеспечивает единый способ отображения на дисплее текстовой информации, ее редактирование, использование в качестве управляющих меню, форматов разного типа, элементов справочной информации (подсказок) и в других целях;

ВИКОНТ — видео-конструктор, предназначенный для интерактивного построения и редактирования графических объектов, композиции сложных графических изображений из более простых элементов с целью их использования в прикладных системах (в том числе, совместно с текстовыми объектами);

ГРИН-1 — графический интерфейс, который служит для интерактивного построения чертежей, схем, карт и других видов изображений с целью их последующего вывода на экран дисплея и на графопостроитель;

АДС — адаптивная диалоговая система, которая включает в себя конструктор и интерпретатор проблемно-ориентированных схем диалога. В основе АДС лежит понятие формальной схемы диалога, описывающей этапы общения пользователя с прикладной системой;

ВИП — виртуальная память, играющая роль базы данных с буферизацией, которая предназначена для хранения системных и проблемно-ориентированных объектов — текстов, структурных данных, внутренних представлений графических объектов и др.

На основе указанных компонентов в ВЦ АН СССР разработан ряд прикладных систем общего и специального назначения.

С экрана дисплея ПЭВМ показана условная схема взаимодействия компонентов интеллектуального интерфейса при построении конкретной прикладной системы (рис. 1). Изображение получено путем применения системы ВИКОНТ. На создание такого изображения затрачивается всего несколько минут.

Имитационная система «Зерно», предназначенная для моделирования сельскохозяйственного производства, позволяет в диалоговом режиме задавать распределение посевных площадей под зерновые культуры и подсчитывать ожидаемый валовый сбор зерна. Эта же система позволяет имитировать динамику уборки урожая по регионам (рис. 2). Окраска каждого региона на схематической карте страны отражает процент собранного урожая. Через каждые 5 секунд, соответствующие одному дню уборки, окраска регионов изменяется в соответствии с цветной шкалой, выведенной в нижней части экрана. Пользователь может изменять и анализировать данные, касающиеся отдельных регионов страны и разных зерновых культур.

Модель «Климат» — это система графической обработки результатов по 3-мерной гидродинамической численной модели климата Земли, разработанной в ВЦ АН СССР. На рис. 3 приведены изолинии изменения среднегодовой температуры воздуха в результате антропогенного изменения климатических параметров. Изолинии получены по данным расчетов на БЭСМ-6, переданных на ПЭВМ по линии связи; они отображаются на экране на фоне упрощенной географической карты Земли. Расчет изолиний осуществляется на ПЭВМ.

Имитационная модель «ГАЗ» (рис. 4, 5, 6) является экспериментальной моделью (на фиктивных цифрах), которая показывает возможности моделирования технико-экономических параметров Единой системы газоснабжения СССР. Пользователь в этой системе имеет дело с распределением добычи, запасов, потребления и транспортировки газа между регионами. Параметры системы могут отображаться в виде гистограмм (рис. 5) или чисел, нанесенных на схематическую карту (рис. 6). Выбор нужной операции обеспечивается текстовыми меню (рис. 4).

Разработаны также другие проблемно-ориентированные системы: РАПО — для имитации распределения сельскохозяйственного производства в районных агропромышленных объединениях; АРМ-ГЕО — для анализа данных геологоразведки; ВИДЕОТЕСТ — для тестирования и психологической тренировки людей с расстройствами высшей нервной деятельности; ряд учебных программ.

Во всех перечисленных системах применяется одинаковая методика организации человеко-машинного взаимодействия. Человек, как правило, пользуется лишь ограниченным набором клавиш или специальным манипулятором для выбора в текстовом или позиционном меню требуемого объекта и запуска обрабатывающей программы. Иногда система запрашивает значения параметров, которые вводятся человеком через алфавитно-цифровую клавиатуру. Общий сценарий работы задается формальной схемой диалога, которая легко подвергается изменениям, позволяя адаптировать систему к конкретным потребностям пользователей.

Реализация интеллектуального интерфейса на основе данного метода позволяет создавать автоматизированные рабочие места разного назначения, отличающиеся удобством использования, возможностью гибкой настройки и в то же время компактностью, надежностью и низкой стоимостью, обусловленной применением персональных ЭВМ.

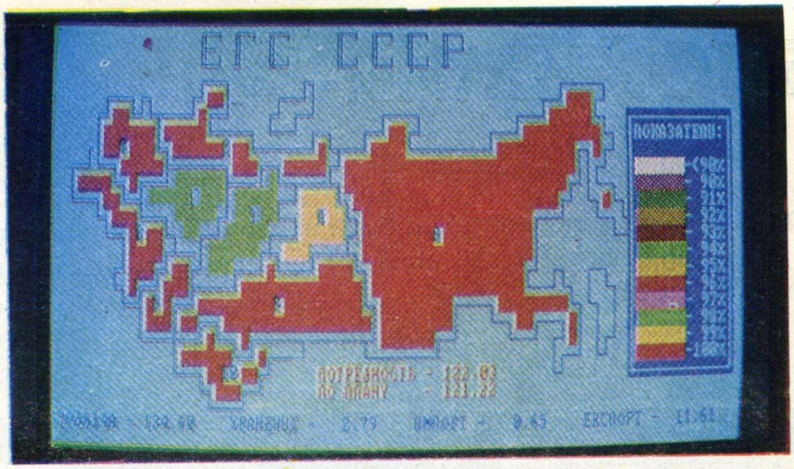


Рис. 4. Система «Газ». Условная карта СССР с отображением показателей ЕГС цветом

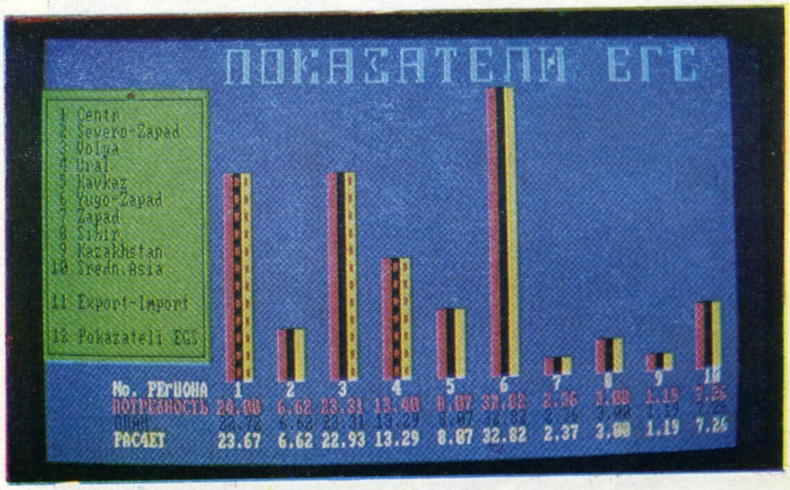


Рис. 5. Система «Газ». Отображение показателей в виде гистограмм

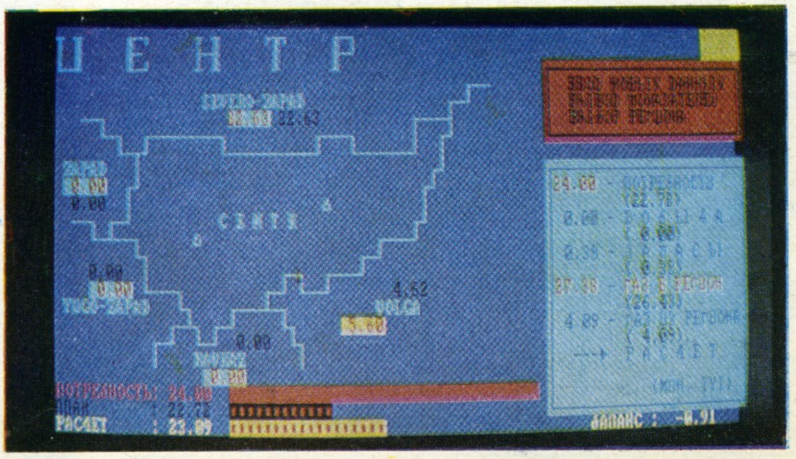
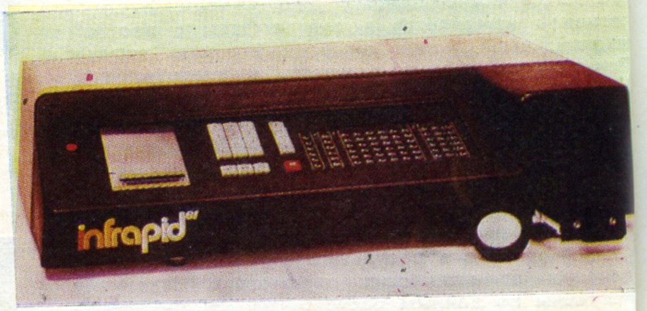


Рис. 6. Система «Газ». Имитация балансовой региональной модели с управлением посредством меню

БИОХИМИЯ-84

С 26 июня по 3 июля в Москве была проведена международная выставка «Биохимия-84» (при XVI конференции федерации европейских биохимических обществ)



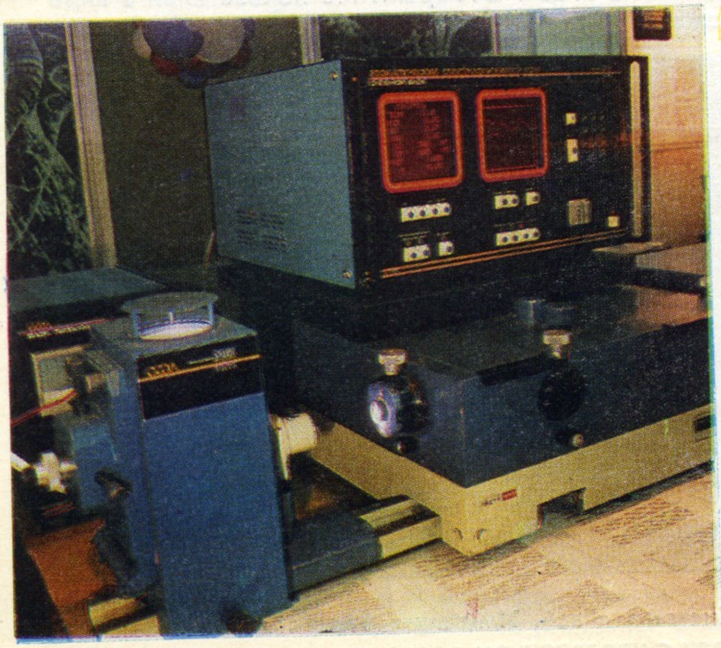


Микропроцессорный экспресс-анализатор кормов Infrapid (ВНР) — фото вверху, справа

Лазерный микрогельхроматограф ХЖ1309 (СССР)— фото вверху, слева

Автоматический спектрофлуориметр с системой управления на основе микроЭВМ «Электроника 60М» (СССР) — фото внизу, слева

Микропроцессорный анализатор количества и размера частиц Coulter ZM (Англия) — фото внизу, справа





каждой ситуации допустимых действий и их последствий. Форма рекомендаций о предпочтении того или иного способа действий. Искусственное затруднение допустимых, но нежелательных для системы действий.

1.4. Соотношение между временем, затрачиваемым на обучение пользователя, и эффективностью его работы с системой. Ориентация системы на сокращение времени обучения или на повышение эффективности работы.

Соотношение между простотой описания системы и простотой ее использования, между теоретическим знанием системы и навыком в ее использовании (даже автору системы приходится некоторое время учиться работе с ней).

1.5. Предоставление пользователю разнообразия функций системы. Всегда ли хорошо включать в одну систему возможность решать много разнообразных задач или лучше предлагать несколько самостоятельных систем? Возможность использования одинаковых или сходных по внешнему виду средств в различных системах.

Всегда ли хорошо предоставлять много разных средств для достижения одной и той же цели (особенно, если нет способа, оптимального во всех отношениях)? Надо иметь в виду, что дополнительные возможности — это также дополнительные требования к пользователю, который должен их знать. Если автор системы считает, что незнание некоторых возможностей не мешает пользоваться остальными, то он не учитывает того, что возможности системы, неизвестные пользователям, с одной стороны, существуют лишь теоретически, не содействуя эффективному функционированию системы, с другой стороны, все равно расходуют определенные ресурсы оборудования.

Должна ли система выглядеть для пользователя законченным объектом, в котором нельзя ничего ни убавить, ни прибавить, или лучше, чтобы она представлялась как расширяемая, имеющая варианты, собираемая в любом комплекте из отдельных функциональных элементов по мере необходимости?

1.6. Особенности работы пользователя с системой, правила которой не полностью известны пользователю. Предсказуемость поведения системы для пользователя.

1.7. Зависимость эффективности работы пользователя от степени его заинтересованности в результате. Может ли та или иная организация системы уменьшить или увеличить степень такой зависимости? (Известно, что многие программисты предпочитают диалоговую систему пакетной обработке главным образом потому, что так они могут исключить незаинтересованного посредника — оператора ЭВМ).

1.8. Методы обследования системы, находящейся в эксплуатации. Оценка эффективности работы пользователей, степени удовлетворенности их системой, выявление типичных затруднений, оценка степени используемости различных средств системы и объяснение различий в используемости. Восприятие системы пользователем. Персонификация системы.

2. Особенности систем разработки и отладки программ

2.1. Программа, очевидно, не сразу возникает в сознании программиста в готовой форме, поэтому можно поставить вопрос о том, что в психике программиста предшествует написанию программы. По оценке Э. В. Дейкстры, промежуточная информация, участвующая в создании программы, по крайней мере вдесятеро превышает объем конечного результата. (Многие квалифицированные программисты склонны скрывать от окружающих промежуточные стадии своей работы, поскольку они не обладают должной законченностью, а часто вообще не осознают этих стадий). В свое время на ответ на поставленный вопрос претендовали «языки высокого уровня», поскольку предполагалось, что они, в отличие от программ в машинном коде, непосредственно представляют исходное намерение программиста. Теперь видно, что это не так. Возможно, пониманию этого вопроса поможет изучение модульности в представлении знаний и разработка соответствующих средств модульности в программировании.

2.2. Поиск наилучших принципов модульной организации программы, требований к межмодульным сопряжениям, варьируемости модулей. Определение оптимального размера модуля.

В обычных системах программирования межмодульные сопряжения гораздо менее эффективны, чем сами модули, что, вместе с традициями старого, безмодульного программирования приводит к существенному завышению размеров модулей. В системе программирования CDL2, где благодаря измененной технологии использования модулей межмодульные сопряжения не снижают эффективности, типичный размер минимальной программной единицы снизился до нескольких строк текста, и этот выбор даже подкреплен небольшими экспериментами.

2.3. Оценка различных подходов к созданию систем из многих модулей. Представление неполностью готовых модулей. Целесообразность автономной отладки (она желательна, однако сама требует усилий на разработку и может служить источником дополнительных ошибок).

Разработка систем методом «снизу вверх» (от модулей, реализующих самые элементарные функции, к основанным на них все более и более содержательным модулям), «сверху вниз» (от общего плана системы к определению функций нижележащих модулей и их реализации), постепенным наращиванием функций уже действующей системы.

2.4. Влияние коллективного характера разработки на модульную структуру создаваемой системы. Степень доступности и понятности модулей другим исполнителям. Закон М. Конвея (модульная структура системы воспроизводит структуру разработавшего ее коллектива).

2.5. Комментирование и документирование программ. Информативность и достоверность комментариев. Привязка комментариев к крупным или мелким единицам программы, ориентация комментариев на описание общих структур и взаимосвязей или на описание отдельных действий. Индивидуальные различия в стиле комментирования. Мотивация комментирования. Роль комментариев в понимании чужой программы.

2.6. Исследование процесса отладки программ. Оценка средств, помогающих отладке.

3. Вопросы, связанные с диалоговыми системами

3.1. Для каких задач предпочтительнее диалоговый режим? (Одни программисты предпочитают вести отладку программ в режиме диалога, другие — в пакетном режиме). Оценка причин, по которым отдается предпочтение диалоговому режиму: быстрота ответа, возможность быстро исправить неточности в запросе, освобождение от составления предварительного плана работы и возможность импровизации.

Нужны ли пошаговые трансляторы с языков высокого уровня? Системы, в которых программа на языке высокого уровня вводится малыми порциями и тут же выполняется, при их кажущейся привлекательности нечасто используются квалифицированными программистами, потому что в их представлении программа состоит из более крупных единиц.

3.2. Общая организация диалога. Строгая последовательность чередующихся вопросов и ответов или относительно независимый поток сообщений с каждой стороны? Выдача пользователем нескольких запросов сразу. Необходима ли формальная привязка ответа к вопросу? Чередование инициативы в диалоге; в каких случаях инициативу должна брать на себя система? (Если диалог организован в виде строгого чередования реплик сторон, то под перехватом инициативы понимается выдача реплики, отличной от нормально ожидаемой на основе предшествующего диалога).

3.3. Информация о текущем состоянии диалога, сохраняемая системой: текущий режим работы, объекты, с которыми в данный момент ведется работа, запросы, еще не получившие ответа, и т. п. Допустимый для пользователя объем такой информации, частота ее изменения. Можно ли допускать изменение этой информации в качестве побочного результата выполнения операций, специально не направленных на такое изменение? Лучше ли изменять эту информацию всю сразу или только по элементам? Целесообразность индикации этой информации по инициативе системы или по запросу пользователя. Какие параметры выполняемых операций лучше задавать неявно (через установку текущего состояния диалога), а какие явно при каждой операции, где они нужны? Что лучше размещать на экране: информацию о состоянии диалога или протокол предыдущей работы? Допустимы ли реплики, понятные лишь при знании текущего состояния?

Нужна ли смена режима при переходе к другому классу операций? Два крайних решения: дерево режимов и подрежимов со своим набором операций в каждой точке и явными переходами по этому дереву или же сплошной набор команд, каждую из которых можно подавать в любой момент независимо от предыдущей работы.

3.4. Контроль работы. В каких случаях контролировать команды пользователя? В каких ситуациях система должна запрашивать дополнительное подтверждение выдаваемой команды? Нужно ли системе реагировать на ошибку в команде, ввод которой не завершен? Подтверждение успешного выполнения команды системой. Дополнительные возможности проверки выполнения системой команды по инициативе пользователя.

3.5. Режим подсказки и обучения пользователя. В каких случаях и по чьей инициативе к нему переходить, как выходить из него? Нужны ли различные уровни такого режима?

3.6. Допустимое время ожидания ответа системы. Допустимые колебания этого времени. Время ожидания системой очередной реплики пользователя. Интервал, позволяющий системе предположить прекращение работы пользователем.

3.7. Процедуры начала и завершения работы. Информация, которую должна сообщать каждая из сторон при этих процедурах.

3.8. Автоматическая регистрация действий пользователя (включая временные характеристики этих действий). Влияние такой регистрации на работу осведомленного о ней пользователя.

3.9. Влияние задержек и сбоев в системе на эффективность работы пользователя. Средства уменьшения такого влияния.

4. Языки взаимодействия пользователя с системой

4.1. Выдача пользователем запроса посредством «меню», т. е. выбора из предложенного перечня вариантов, или посредством заполнения большой таблицы, предусматривающей значения всех возможных параметров, или же в виде компактных команд, подчиненных определенным синтаксическим правилам. Допустимая сложность синтаксиса команды. Совмещение нескольких действий в одной команде. Совмещение нескольких объектов в одной команде. Соотношение между длиной команды и ее информационной нагруженностью.

Языки, требующие представления предложений в виде стандартной последовательности полей с размещением этих полей в определенных колонках бланка, и языки со свободным форматом текста. Правила переноса длинных предложений в языках с поколонным форматом.

Использование «по умолчанию» стандартных значений некоторых параметров при отсутствии явного их задания. При программировании на языке ПЛ-1, где использование аппарата умолчаний доведено до абсурда, часто возникают трудно обнаруживаемые ошибки из-за того, что система подставляет весьма неожиданные с точки зрения пользователя умалчиваемые значения.

4.2. Выбор длинных или коротких обозначений, длинных или коротких способов записи конструкций языка. Соотношение между легкостью записи и легкостью понимания текста другим человеком. Упрощается ли понимание при длинном написании? Использование синонимических вариантов, синонимия длинного и краткого написания. Выбор функции, которая будет присвоена команде минимальной длины (в диалоговых системах — нажатию на клавишу типа «внимание»; некоторые пользователи склонны нажимать на такую клавишу в случае задержек так же, как иногда стучат по рычагу телефонного аппарата).

4.3. Совмещение разных языков в одной системе. Различие между языком сообщений системы и языком сообщений пользователя (как правило, они различны). Необходимая степень сходства между ними. Использование разных языков для разных категорий пользователей. Параллельное использование текстовых языков, функциональных клавиш и графических средств. Распределение функций между ними.

КОММЕНТИРУЕТ О. К. ТИХОМИРОВ,

доктор психологических наук, профессор МГУ

Сейчас все большее число специалистов понимает, что использование ЭВМ ведет к существенному повышению производительности труда. Однако создание ЭВМ и их программного обеспечения — это тоже труд, характеризуемый своими показателями эффективности. Повышение этих показателей актуально в период, когда программным обеспечением заняты все новые и новые группы специалистов. Хорошо известно, что психологические факторы существенно влияют на эффективность любого труда. Это, очевидно, относится и к работе по созданию программного обеспечения.

Полностью можно согласиться с главным положением статьи (или точнее заметок) Г. С. Цейтина: «... организация взаимодействия с пользователем выдвигается на первое место». Согласен, что основные трудности состоят «в том, чтобы решить, какое именно поведение (ЭВМ) для нас желательно». Г. С. Цейтин совершенно прав, отмечая, что психологические проблемы изучения и проектирования деятельности пользователей ЭВМ существенно отличаются от тех, которыми занимается инженерная психология.

Г. С. Цейтиным составлена классификация психологических задач, возникающих в ходе разработки программного обеспечения. Всего таких задач выделено 37, они объединены в шесть классов, причем некоторые из них включают в себя множество подзадач (например, «1.2»). Конечно, предложенный список проблем нельзя считать полным, основания для классификации не являются строгими, но сама работа по составлению такого предварительного списка безусловно полезна как для разработчиков программного обеспечения, так и для профессиональных психологов. Задачи формулируются в виде вопросов, на которые нужно получить ответ, зависимостей, которые должны быть выявлены, и параметров, которые должны быть проанализированы. Кроме перечня задач, Г. С. Цейтин приводит ряд интересных наблюдений и формулирует некоторые обобщения: от причин предпочтения диалогового режима до влияния вечерних часов работы программистов на их женитьбу на девушках-операторах.

Вместе с тем решительно нельзя согласиться с мнением Г. С. Цейтина о том, что «психология ... сегодня не только не может выдать конкретные практические рекомендации, но даже не предлагает общей теоретической ос-

4.4. Использование естественного языка или его элементов.

Идея использования во взаимодействии с машиной естественного языка (родного языка пользователя) выглядит очень привлекательно, однако для большинства создаваемых сегодня систем она не оправдывает себя, и не столько из-за трудностей автоматического анализа сообщений пользователя, выраженных на естественном языке в произвольной форме, сколько из-за того, что в пределах жестко ограниченной ситуации, связанной с функционированием автоматической системы, использование всего богатства средств естественного языка выглядит н е е с т е с т в е н н о; сам естественный язык в подобных ситуациях имеет тенденцию к сжатию, сокращению словаря, упрощению синтаксиса вплоть до формализации (например, в речи диспетчеров). Предложение обращаться к системе на естественном языке может ввести пользователя в заблуждение относительно действительных возможностей системы и привести к появлению запросов, невыполнимость которых будет трудно объяснить. Вероятно, вопрос об использовании естественного языка можно будет поставить более серьезно, когда один пользователь будет взаимодействовать с большим количеством разнообразных систем, неизвестных ему в деталях, и вместо многих частных языков потребуется объединяющий язык. На сегодняшний день можно говорить только об использовании элементов естественного языка.

Возможные частные вопросы: исследование спонтанных ограничений и структурных изменений естественного языка при использовании его в жестко ограниченных ситуациях; исследование интерференции формальных языков при параллельной работе пользователя с различными системами (что лучше: частичное сходство или полное несходство?); выбор имен для искусственно вводимых объектов и операций (степень сходства со словами обычного языка, предпочтение коротких или длинных имен, использование сокращений, допустимость метафор, создание искусственных систем именования, именование при невозможности использования алфавита родного языка); целесообразность использования в ответах системы развернутых текстов на естественном языке, получаемых заполнением шаблонов.

5. Ввод и вывод текстовой информации

5.1. Работа на клавиатуре. Конструкция клавиатуры и размещение знаков. Совмещение нескольких клавиатур (например, русской и латинской: на одну клавишу помещать буквы, сходные по звучанию, или сохранять для каждой из клавиатур стандартное расположение?). Переключение регистров: постоянный переход на другой регистр или нажатие клавиши регистра одновременно со знаковой клавишей? Должен ли пользователь помнить текущий регистр? Удобно ли пользователю выбирать один «основной» регистр и всегда возвращаться к нему, переходя на другой регистр только ради конкретных знаков? Можно ли системе переключать регистры клавиатуры? Трудности работы с многими регистрами (например, два русских, два латинских и управляющие знаки); задавать переключение регистров: переход на (1) русские (2) строчные буквы — две клавиши или одна? Работа с курсором; как выводить его в нужную позицию?

Типичные ошибки при работе на клавиатуре. Должна ли система стремиться исправлять их? Способы исправления ошибок во вводе, замеченных самим пользователем.

Индикация информации, вводимой с клавиатуры. Непосредственная индикация, задержанное подтверждение со стороны системы (например, индикация после окончания строки или после завершения обработки введенного символа), работа без индикации (вслепую). Передача данных в машину при каждом нажатии клавиши или только по специальной «исполнительной» клавише.

5.2. Работа с неполным алфавитом (например, при отсутствии букв русского алфавита, отсутствии строчных букв или специальных знаков). Транслитерация отсутствующих знаков, обратимость транслитерации; контекст, необходимый для восстановления исходной записи по транслитерации. Использование авторегистра (escape character). Использование обычной клавиатуры для подготовки многошрифтового типографского набора.

При разработке языка с большим алфавитом лучше предусматривать транслитерацию в ограниченный алфавит или для каждой конструкции указывать дополнительный способ представления при условии ограниченного алфавита?

При совместном использовании русского и латинского алфавита отождествлять ли графически сходные буквы разных алфавитов? Если различать, то какими средствами и какой контекст использовать для различения (например, принадлеж-

новы, в рамках которой можно было бы выделять конкретные исследовательские задачи и ставить эксперименты». Парадокс заключается в том, что автор анализирует крайне ограниченный массив психологических исследований, учитывает не все работы.

Приведу лишь некоторые примеры. Задача «3.6» (допустимое время ответа системы) разрабатывается во многих исследованиях и была предметом специального аналитического обзора, не упоминаемого автором. Имеются специальные обзоры проблем, возникающих в практике программного обеспечения ЭВМ, при автоматизации проектирования, управления, научных исследований и обучения. Все они не учтены автором. Выделены параметры психологического анализа и оптимизации диалога человека—ЭВМ.

Все эти развернутые в настоящее время исследования имеют общую теоретическую основу: научные психологические знания о природе творческой деятельности, общения, личности.

Не вполне ясным представляется положение Г. С. Цейтина о том, что «психология не встречалась прежде с задачами этого класса». На самом деле исследования психологических факторов, влияющих на эффективность взаимодействия с ЭВМ, ведутся уже четверть века.

Тем не менее, приходится признать, что большую часть психологических задач, уже возникших и вновь непрерывно возникающих в ходе разработки программного обеспечения, еще предстоит решить. Чтобы эта работа была более эффективной, необходимы, по крайней мере, три условия:

а) база для эмпирических исследований, проверки некоторых априорных гипотез и предлагаемых рекомендаций;

б) психологическая готовность разработчиков программного обеспечения к кооперации с психологами (а не только к их критике);

в) организация этой совместной работы.

В настоящее время психологические задачи, возникающие в практике разработки программного обеспечения, решаются сравнительно небольшими исследовательскими группами. Между тем только сформулированных Г. С. Цейтиным 37 задач хватило бы для интенсивной работы целого института на ближайшие пять лет! Таким образом, наряду с необходимостью выделения психологических задач в разработке программного обеспечения возникает необходимость программирования работы по решению этих и аналогичных задач.

В целом выделение и решение психологических задач в разработке программного обеспечения представляется мне не менее важной и интересной работой, чем сама разработка программного обеспечения. Это — перспективное в научном отношении направление.

ность соседних букв к тому или другому алфавиту)? Если не различать, то как устанавливать соответствие между строчными и прописными буквами?

5.3. Размещение текстового материала на экране или на печати. Способы группировки информации. Использование пробелов. Должно ли играть роль число подряд идущих пробелов? Способы выделения на экране: яркость, цвет, мерцание, «негатив», переход на другой регистр, отступ от левого края и т. п. Способ различения в протоколе диалога реплик системы и пользователя. Нужно ли менять содержимое экрана все сразу или постепенно? Варианты: продвижение старой информации вверх с появлением новой информации внизу, деление экрана на относительно постоянные зоны с фиксированными функциями, так что каждый раз меняется только одно из полей. «Виртуальные экраны», представляемые на реальном экране поочередно или совместно, с частичным перекрытием.

Изменение информации, индицируемой на экране, явной командой или непосредственным исправлением содержимого экрана (экранный редактор).

В операционной системе ОС ЕС перечень команд, управляющих размещением информации на экране консоли оператора, состоит из нескольких десятков строк.

5.4. Объем информации на экране и скорость ее восприятия. Допустимая скорость смены содержимого экрана. Восприятие изменений информации на экране (полезно ли сопровождать изменения звуковым сигналом?). Измерять ли объем информации площадью экрана, количеством букв или количеством более крупных единиц (слов, «предложений»)? Лучше ли показывать много информации сразу, выдавать ее последовательно по частям или выдавать более подробную информацию только по запросу? Удобно ли одному пользователю иметь несколько реальных экранов?

6. Некоторые вопросы социологического характера

6.1. Ограничение прав пользователей и их взаимная защита. Совмещение разграничения доступа к ресурсам и возможности сотрудничества между пользователями (включая выполнение работы по поручению другого пользователя). Слежение за работой пользователей при помощи системы.

6.2. Количественные ограничения использования ресурсов. Ведет ли жесткое нормирование к тенденции избыточного расходования ресурсов? Лучше ли распределять ресурсы между пользователями равномерно по времени или предоставлять их поочередно разным пользователям? Во втором случае, с одной стороны, пользователи, не имеющие ресурса в данный момент, могут переключиться на другую работу, с другой стороны, когда время предоставления ресурса близко к концу, есть тенденция обращаться к машине без достаточного основания вместо того, чтобы спокойно обдумать уже полученные результаты.

6.3. Передача прав на определенный ресурс другому пользователю. Требуется ли для этого разрешение руководства? Включает ли передача права на ресурс право его дальнейшей передачи? Включает ли передача права передачу ответственности (например, зачет ресурса в лимит нового хозяина)? Передача ресурса в частичное пользование.

6.4. Привилегии в доступе к ресурсам. Соотношение с уровнем квалификации пользователя и с его положением. Должны ли привилегии служить защите интересов определенных пользователей или повышению общей эффективности системы? Влияние уровня привилегий на отношение пользователя к работе в системе. Может ли передаваться привилегия, состоящая в праве установления привилегий?

6.5. Способы подтверждения личности пользователя перед системой. Выбор паролей. Равновесие между запоминаемостью пароля и его секретностью.

6.6. Разнообразные конкретные вопросы и случаи.

Если в системе существует электронная почта для пользователей, то нужно ли сигнализировать ошибку при посылке сообщения несуществующему пользователю? Если такая сигнализация будет, то кто угодно сможет этим воспользоваться, чтобы выяснить, состоит ли определенное лицо в списке пользователей системы, а это не всегда желательно.

Некоторая система может обслуживать в режиме диалога многих пользователей, но только поочередно (нормальный цикл работы с одним пользователем — несколько минут). Оказалось, что некоторые пользователи прерывают работу посередине, не освобождая системы, и тогда никто не может ею пользоваться. Как усовершенствовать систему? Если предписывать пользователю ограниченное время для ответа системе, это вызовет у него дополнительное напряжение; если пользователю, обратившемуся к занятой системе, сообщать, кем она занята, это может способствовать возникновению личных конфликтов. Приемлемо ли такое решение: отсчитывать время, но только от момента неудачного обращения к системе со стороны другого пользователя?

Физик, сдающий в вычислительный центр свои задачи для пропуска оператором в пакетном режиме (преимущественно в ночное время), произвел следующий эксперимент: сдал две заявки с идентичными задачами, указав в них разное ожидаемое время счета — 10 мин и 2 ч (реальная потребность — несколько минут). Была выполнена в первую очередь двухчасовая заявка. Причина: оператор предпочел длинную задачу, чтобы за время ее прохождения на машине спокойно поспать.

Ответ на дискуссии об «электронной свахе»: программисты, работающие преимущественно в вечерние часы, часто женятся на девушках-операторах.

Статья поступила 11 мая 1984 г.

МИКРОЭЛЕКТРОНИКА ВО ФРАНЦИИ

За последнее десятилетие стоимость микроэлектронных компонентов уменьшилась в 100 раз и продолжает сокращаться ежегодно на 20—30%. Это позволяет расширить использование микроэлектроники не только в производстве и услугах, но и в быту. По мнению Комиссариата по планированию (Франция), под воздействием новейшей техники быстро будут развиваться следующие сферы:

— робототехника;

— оборудование легковых автомашин. В 1980 г. стоимость электронных компонентов в автомобиле составляла во Франции 40 франков, в Японии — 500. К 1985 г. она возрастет соответственно до 150 и 700 франков;

— бытовая техника. К 1985 г. с помощью микрокомпьютеров будут управляться примерно 35% бытовых приборов в Западной Европе;

— медицинское электронное оборудование и инструменты;

— обучающие и развивающие игры и игрушки, в том числе и для взрослых;

— конторское оборудование.

Согласно исследованию, проведенному во Франции, средства автоматизации к 1985 г. позволят сократить 210 тыс. рабочих мест, в том числе 82 тыс. — на машинописных работах. Одновременно будут созданы новые рабочие места, в частности в области сбора, обработки и распространения информации на основе новейшей техники и технологии.

«ЭКО». РЖ «Общественные науки за рубежом». Серия 2, «Экономика», 1983, № 2.

УДК 681.3.06

Е. А. Иванов, Л. Л. Муренко, Ю. Ф. Широков

УНИВЕРСАЛЬНАЯ ОТЛАДОЧНАЯ СИСТЕМА АВТОМАТИЗАЦИИ ПРОЕКТИРОВАНИЯ МИКРОПРОЦЕССОРНЫХ УСТРОЙСТВ

Для разработки изделий на основе микропроцессоров требуются средства проектирования и отладки, позволяющие выбирать самое оптимальное из возможных структурных, архитектурных и схемотехнических решений.

Предлагаемая *универсальная отладочная система* «Электроника Т3» обеспечивает возможность построения отладочных средств для различных типов микропроцессоров с фиксированной системой команд и предусматривает быструю адаптацию применительно к новым, более мощным микропроцессорам и однокристальным микроЭВМ. Это достигается благодаря максимальной унификации аппаратных и программных средств системы по интерфейсам и системе команд, выбору в качестве базового вычислительного комплекса (БВК) серийно выпускаемых вычислительных средств с системой команд микроЭВМ «Электроника 60» (15 ВУМС-028-025, ДВК «Электроника НЦ-80—20/2) и программной реализации основных функций.

Процесс проектирования микропроцессорных устройств (МПУ) включает три основных этапа: разработка, отладка и комплексная стыковка аппаратных и программных средств. Отладочные средства на всех этапах проектирования МПУ должны предусматривать прогон программы пользователя с произвольного адреса без прерываний и с остановом в контрольных точках; пошаговое выполнение программы; изменение и отображение на дисплее содержимого памяти и рабочих регистров микропроцессора (МП). *Кроме* выполнения прямых функций *отладки* система «Электроника Т3» *может использоваться* в качестве инструментальной *для разработки* программного обеспечения, тестирования и испытания.

По выполняемым функциям и составу аппаратного и программного обеспечения отладочные средства можно разделить на три основных класса: устройства, комплексы и программные кросс-средства. В каждом классе отладочные средства могут быть специализированными — для одного МП, и универсальными — для группы МП.

Отладочная система «Электроника Т3» включает следующие аппаратные и программные модули (рис. 1):

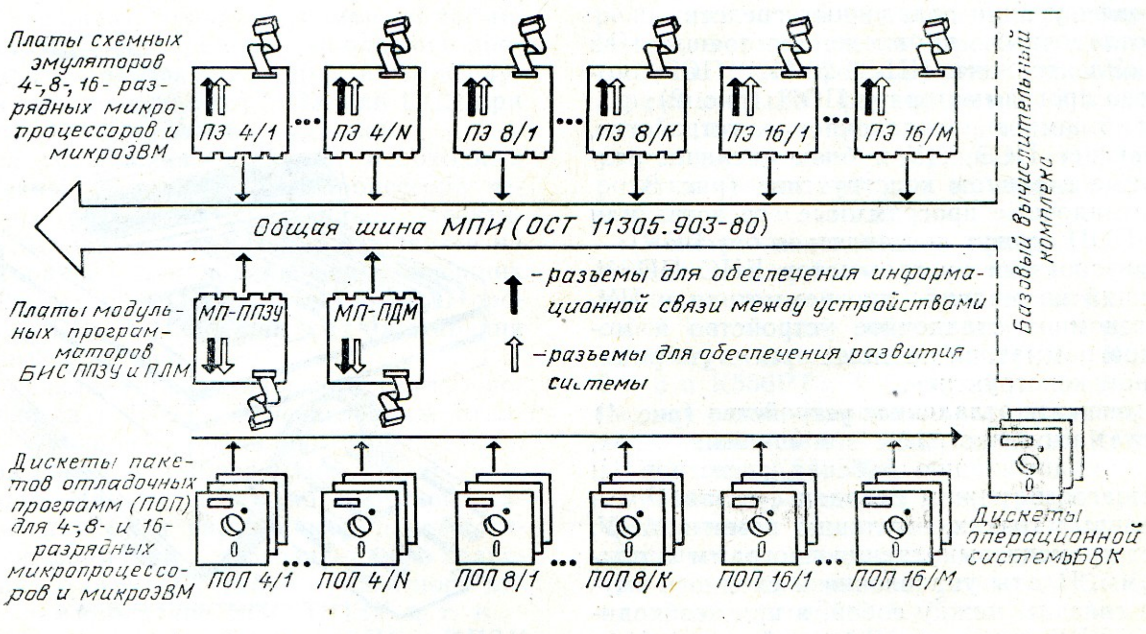


Рис. 1. Состав аппаратных и программных модулей отладочной системы «Электроника Т3»

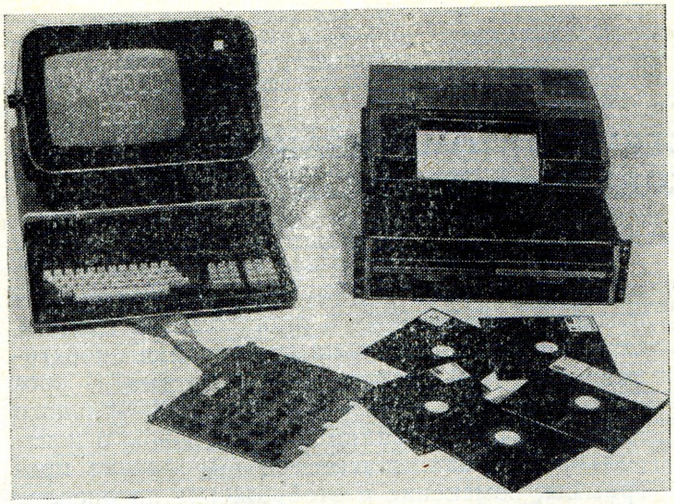


Рис. 2. Отладочный комплекс на основе ДВК «Электроника НЦ-80—20/2»

— набор плат схемных эмуляторов (ПСЭ), число которых должно соответствовать числу типов выпускаемых 4-, 8- и 16-разрядных микропроцессоров;

— плату управления (ПУ) с микроЭВМ, однострочным дисплеем и клавиатурой;

— две платы модулей программаторов БИС ППЗУ и БИС ПЛМ;

— пакеты отладочных программ (ПОП) на гибких магнитных дисках, соответствующие числу основных типов выпускаемых микропроцессоров.

Аппаратные и программные модули отладочной системы позволяют создавать следующие конфигурации отладочных средств:

— отладочный комплекс, состоящий из ДВК «Электроника НЦ-80-20/2», ПСЭ, модульного программатора и ПОП (рис. 2);

— автономное отладочное устройство, включающее ПСЭ, ПУ и блок питания, размещенные в едином конструктиве (рис. 3);

— отладочные программные средства или набор ПОП (могут поставляться без БВК);

— автономный программатор БИС ППЗУ, состоящий из модуля программатора и ПУ.

Автономное отладочное устройство и модуль программатора размещаются в унифицированном конструктиве.

Автономное отладочное устройство (рис. 4) реализует ввод программ в машинных кодах, опрос и модификацию рабочих регистров и ЗУ, пошаговый режим отладки с остановом в контрольных точках, эмуляцию памяти МПУ, подсчет времени выполнения программ и подпрограмм. Платы управления и схемного эмулятора связаны между собой, а при необходимости с БВК через межплатный интерфейс (МПИ). В состав ПУ входят однострочный дисплей, на который выводится адрес команды, код команды и данные, 20-клавишная клавиатура для ввода программ, данных и управления режимами работы ПЭ, ОЗУ для хранения программ и данных объемом 2 Кбайт, ПЗУ для хранения программы-монитора, МП К580ИК80 с тактовым генератором и адаптер связи с МПИ. Связь отдельных функциональных узлов ПУ осуществляется через шину адреса, данных и управления МП К580ИК80.

ПСЭ содержит в своем составе БИС МП, аналогичную используемой в проектирумом МПУ, тактовый генератор, ОЗУ для хранения отлаживаемой программы, ПЗУ для хранения управляющих программ, ЗУ контрольных точек останова, компаратор, схему управления циклом эмуляции МП, карту памяти, таймер, регистр команд и данных, адаптер МПИ, схему управления шинами эмулятора и МПУ, шинный формирователь (с помощью гибкого кабеля и штыревого разъема соединяется с розеткой БИС микропроцессора МПУ).

Управление схемным эмулятором, а также информационный обмен между его памятью и микроЭВМ осуществляется с помощью четырех регистров команд управления и данных с фиксированными адресами, задаваемыми на ПСЭ с помощью перемычек. Первые два регистра используются в качестве регистров адреса и данных памяти, а вторые — в качестве регистров команд и статуса.

ЗУ контрольных точек, водимых с ПУ или через адаптер МПИ из БВК, компаратор и

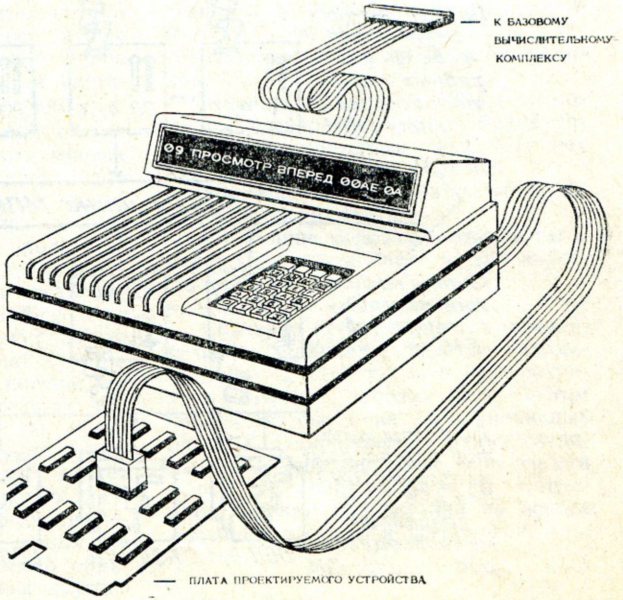


Рис. 3. Автономное отладочное устройство

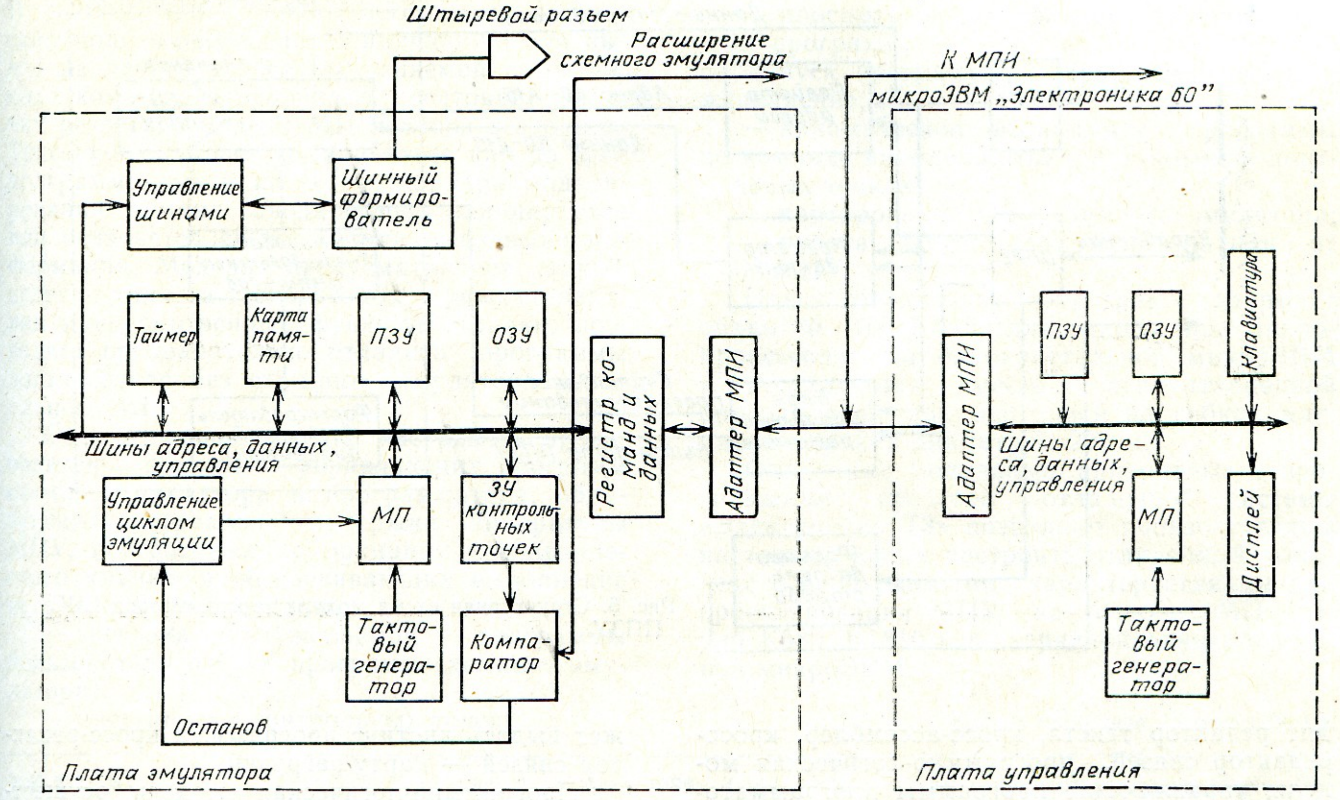


Рис. 4. Структурная схема автономного отладочного устройства

схема управления циклом эмуляции позволяют реализовать пошаговый режим отслеживания за состоянием шины эмулятора с остановом в контрольных точках. С помощью таймера ведется подсчет времени выполнения программ или подпрограмм. Карта памяти служит для эмуляции памяти МПУ или ПСЭ при диагностике ошибок, сбоев и отказов.

Схема управления шинами необходима для организации связи микропроцессора через шинные формирователи с шиной проектируемого МПУ. Шина эмулятора выводится на внешний разъем для подключения внешних периферийных устройств (кассетное ЗУ, программатор БИС ППЗУ) или для расширения отладочных функций.

При подключении автономного отладочного устройства к БВК к перечисленным отладочным функциям добавляются ввод программ на языке ассемблера микропроцессора, трассировка (до 64 состояний адресной шины, шины данных и управления), документирование информации, долговременное хранение и ввод программ с гибкого магнитного диска, программирование БИС ППЗУ и т. п.

Модуль программатора БИС ППЗУ (рис. 5) содержит следующие функциональные узлы: шинные формирователи ШФ1 и ШФ2 для связи с МПИ, 16-разрядные регистры адреса и данных ППЗУ, блок управления режимами и выбора типа программируемого ППЗУ, формирователь строба пассивного устройства СИП, формирователь временной диаграммы ППЗУ, формирователь высоковольтных импульсов программирования ППЗУ с ультрафиолетовым стиранием информации, розетка соединений для вставления программируемой БИС ППЗУ.

Конструктивно модуль программатора выполнен в виде блока управления и контактирующего устройства, соединенных между собой с помощью гибкого жгута, и рассчитан на программирование БИС ППЗУ К573РФ1, 2 и 3 и К556РТ3, 4, 5, объемом до 2^{16} 16-разрядных слов. Максимальное время программирования — 4 мин для одной 16 Кбит БИС ППЗУ типа К573РФ2.

Пакет отладочных программ (рис. 6) предназначен для отладки как *программного обеспечения* микропроцессорных устройств при использовании совместно с БВК, так и *аппаратных средств МПУ* на отладочном комплексе в составе БВК, схемного эмулятора и программатора ППЗУ. В состав пакета вхо-

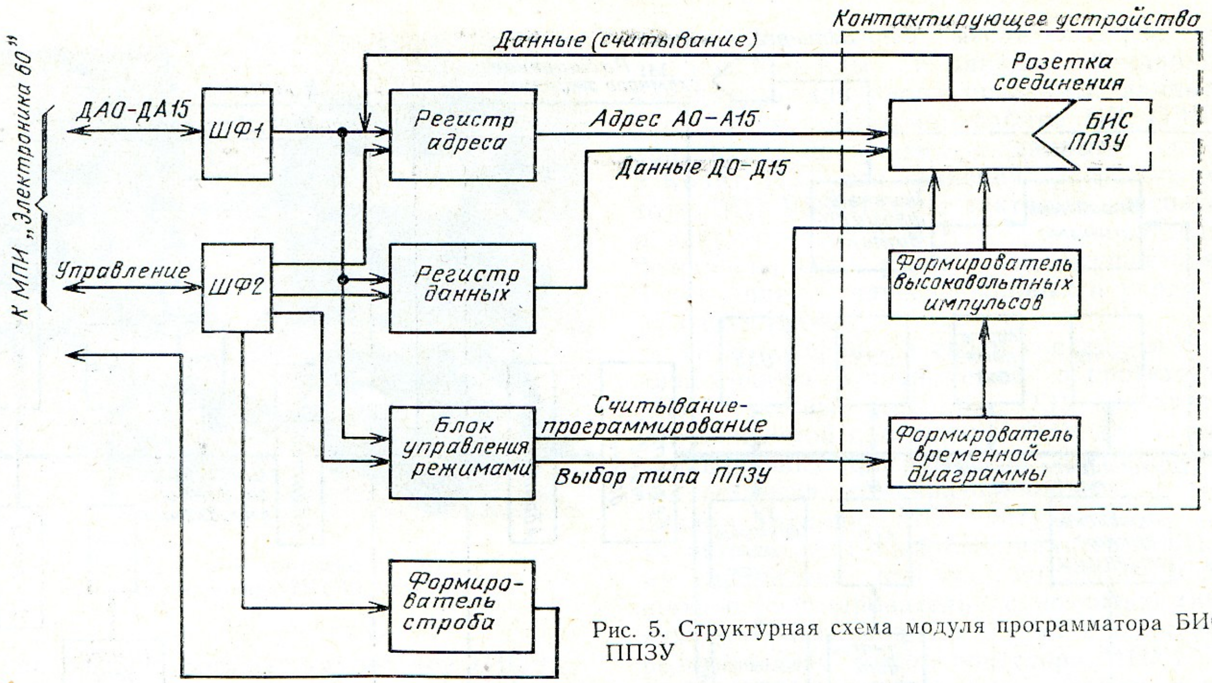


Рис. 5. Структурная схема модуля программатора БИС ППЗУ

дят редактор текста, кросс-ассемблер, кросс-редактор связей, программно-логическая модель, управляющие программы программатора и схемного эмулятора.

Кросс-ассемблер формирует объектный модуль, который преобразуется в загрузочный с помощью кросс-редактора связей. В случае необходимости кросс-ассемблер может выдать листинг программ, а кросс-редактор связей — карту загрузки.

Программно-логическая модель осуществляет покомандную интерпретацию (моделирование) загрузочного модуля с целью проверки выполнения. В случае необходимости модель может выдать листинг программы в кодах, а также справочную информацию.

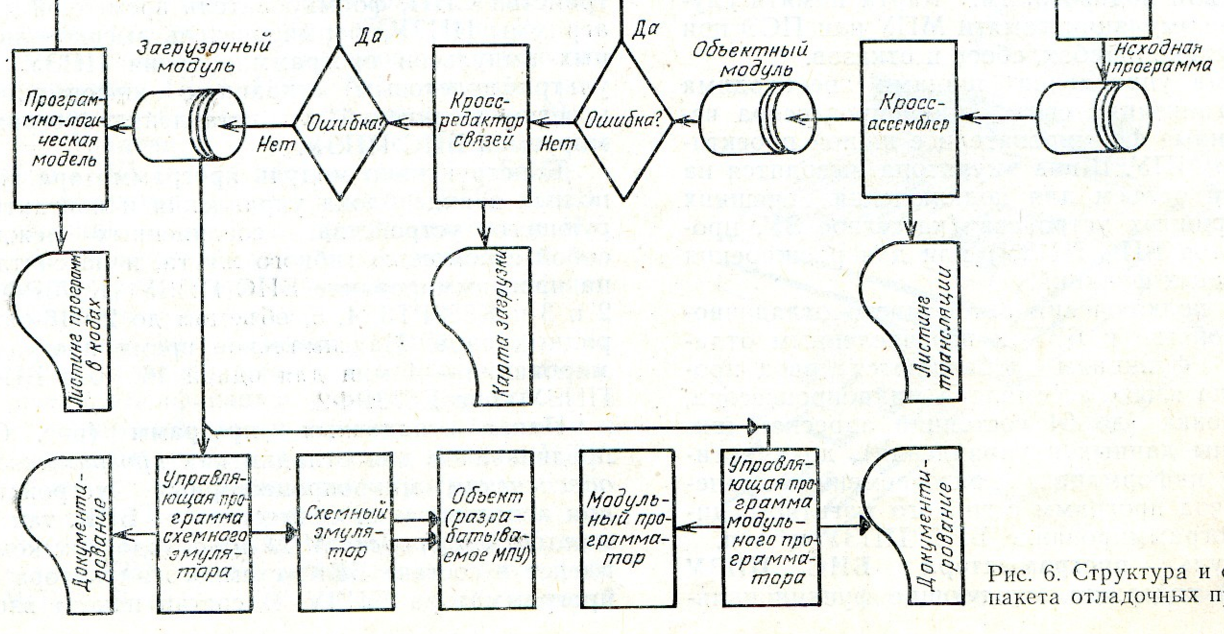


Рис. 6. Структура и состав пакета отладочных программ

В режиме трассировки программно-логическая модель обеспечивает индикацию или вывод на печать адреса выполняемой команды, кода команды и операндов, состояние рабочих регистров модели МП и слова состояния. Трассировка может выполняться как по всей программе, так и по ее части. При моделировании команд ввода-вывода индицируется или печатается вводимая или выводимая информация. Модель предусматривает также ведение диалога с оператором в режиме задания предусмотренных функций отладки, позволяет предварительно отладить программы пользователя без схемного эмулятора и реального МПУ.

Окончательная отладка программы производится с помощью управляющих программ схемного эмулятора и программатора, образующих программы поддержки аппаратуры МПУ, построенных по принципу иерархического «меню» и обеспечивающих выполнение всех отладочных функций:

— загрузка отлаживаемой программы пользователя в ОЗУ программ схемного эмулятора;

— расстановка контрольных точек;

— просмотр и модификация содержимого ОЗУ программ;

— пуск программы, останов и пошаговое выполнение;

— динамическое выполнение программы по команде «продолжить» с остановом в контрольных точках;

— картирование ОЗУ схемного эмулятора;

— работа МП от ОЗУ эмулятора или от ОЗУ МПУ.

Пакет отладочных программ занимает объем 80—100 Кбайт для одного типа микропроцессора или однокристальной микроЭВМ и поставляется вместе с эксплуатационной программной документацией на гибких магнитных дисках диаметром 203 мм.

Описанные модульные аппаратные и программные средства отладочной системы «Электроника Т3» позволяют пользователю в настоящее время построить отладочный комплекс с необходимыми характеристиками для проектирования МПУ на основе МП серий К580, К1810 и 8-разрядной однокристальной микроЭВМ.

Статья поступила 23 марта 1984 г.

РЖ ВИНИТИ

1Б378. [МикроЭВМ с аппаратным интерпретатором языка BASIC]. Катода Хироси, Огата Микио, Нисино Нобуо. «Дэнси дзайрё, Electron. Parts and Mater.», 1983, 22, № 4 (яп.)

Рассматривается карманная микроЭВМ, содержащая две БИС μРД7904/7905) интерпретатора с языка BASIC. Приводятся состав команд, структура и основные характеристики языка BASIC, а также технические характеристики и перечень БИС, входящих в состав карманной микроЭВМ.

1Б1. Социальные потребности и последствия создания ЭВМ 5-го поколения. What is required of the 5th generation computer social needs and its impact. Karatsu Hajime. «5 Generat. Comput. Syst. Proc. Int. Conf., Tokyo, Oct. 19—22, 1981». Amsterdam e. a., 1982 (англ.)

Предполагается, что в результате создания ЭВМ 5-го поколения в Японии появится новый тип инфраструктуры — информационный, будут ликвидированы профессии, связанные с тяжелым физическим трудом, значительно увеличится производительность труда в отраслях с традиционно низким его уровнем: в образовании, медицине, административном управлении.

Прошлое, настоящее и будущее систем управления с ЭВМ. The computer and control yesterday, today, and tomorrow?. Docherty P. P., Nunns S. R. «4th Int. Conf. Trends On-Line Comput. Conf. Syst., Warwick, 5—8 Apr., 1982». London; New York, 1982 (англ.).

Характеризуются технические структуры АСУ ТП и других управляющих систем реального времени, разработанных в 1950—80-х гг. Обсуждаются их особенности и возможности. Отмечается большое влияние МП-средств на эволюцию структур АСУ ТП. Среди современных децентрализованных АСУ ТП выделяются системы, построенные на принципе локальной сети с кольцевой системой связи.

В качестве определяющих тенденций развития подобных управляющих систем рассматриваются: широкое использование заказных БИС, дающих возможность аппаратной реализации ряда функций, выполняемых программным способом (в том числе операционных систем); перспективность языка программирования ADA; безусловное использование децентрализованного принципа построения.

ТРУДНОСТИ ВНЕДРЕНИЯ ЭВМ В УПРАВЛЕНИЕ

Бытует мнение, что вычислительная техника сама наведет порядок в информационных системах. Однако с каждым годом все больше фирм сталкиваются со значительными трудностями при эксплуатации ЭВМ. На первый взгляд, золотой век компьютеров — в разгаре. К примеру, в США вычислительная мощь установленных ЭВМ возрастает на 30% в год. Спрос на них не снижается, сфера применения расширяется. Положительный эффект использования электронно-вычислительных машин несомненен. Но ...

Затраты на математическое обеспечение растут быстрее, чем снижаются издержки на аппаратуру. Если до 1980 г. общая стоимость информационных систем падала, то, по прогнозам, к 1985 г. она вновь поднимется до уровня 1975 г. Резко возрастет и спрос на программистов: с 1980 по 1990 г. — на 43%.

Большинство фирм не поспевают за быстрыми изменениями в вычислительной технике. Порой, махнув рукой на излишне быстрый научно-технический прогресс, компании продолжают эксплуатировать ЭВМ 60-х и 70-х годов. Несмотря на продолжительную историю внедрения вычислительной техники, многие корпорации недостаточно опытны в работе с ней. Они постоянно прибегают к помощи консультантов.

Сфера применения ЭВМ расширяется настолько быстро, что программисты порой не готовы к решению новых задач. Внедрение многих систем управления растягивается на годы. За это время проект устаревает, однако подкорректировать его мешает неповоротливость системы управления. Обычно внедрение ЭВМ порождает многочисленные комиссии, которые только усложняют и без того запутанную систему управления.

«ЭКО». «Harvard Business Review» (США), 1982, N 11—12.

УДК 681.3.06

Л. В. Городняя, А. Н. Кирпотин

НЕКОТОРЫЕ ОСОБЕННОСТИ ПОВСЕДНЕВНОЙ ПРАКТИКИ ПРОГРАММИСТА

Присматриваясь к программированию, заинтересованная в этой молодой профессии общественность породила героический образ «суперпрограммиста», который, подобно фокуснику, неведомо откуда берет программы и наделяет ими всех. Среди самих программистов бытует очень похожий образ идеального программиста «экстра-класса», полученный экстраполяцией дарований, способствующих производительности в программировании. Он и общительный, и обаятельный, и все знает, и все понимает, и всему научается, и любая ЭВМ ему подчиняется (только зачем он и днюет, и ночует у ЭВМ, да на семью и сослуживцев жалуется: мол, от работы отвлекают?).

Интересно, что большинство теоретических исследований в области программирования несут на себе отпечатки именно героического образа, постулируя существование правильных программ, спецификаций, доказательств и других сложных построений, хотя в фольклоре программистов издавна отражена убежденность в малореальности таких предпосылок: «в каждой работающей программе есть хотя бы одна ошибка». Впрочем, в настоящее время наблюдается резкое оживление интереса к задержанным данным, незавершенным процессам, неполным определениям и вообще нечетким понятиям. Полезный, возможно, как цель в процессе самовоспитания программистов, этот образ «суперпрограммиста» не способствует правильному пониманию программирования как профессии. Любой здравомыслящий человек возмущенно воскликнет: «Да полноте! Я знаком с двумя программистами. Ничего особенного в них нет. Они такие же, как все».

И вот, с легкой руки Э. Дейкстры, на сцену скромно выходит «смиренный программист». Его научили кое-чему, и что-то он сам нашел или понял. У него семья, дети, общественные нагрузки, работает он урывками, балансируя между личной жизнью и ЭВМ, судорожно исправляя ошибки, забывая иногда очень важные вещи и с трудом наверстывая упущенное. Этот программист сознает, что не все он умеет, его способности ограничены, времени не хватает, и машина ломается, и свою задачу он не вполне знает, а когда, наконец, в точности все понял и сумел (программа готова, больше задачу программировать незачем), появится новая задача, которая не вполне понятна.

Можно сделать вывод, что **работа программиста заключается именно в «уточнении знания» задачи и представлении уточненного знания в виде текста программы, выполнение которой на ЭВМ является решением задачи.** Успешность этой работы обеспечивается специальной организацией труда, основанной на понимании скромности своих средств. Но будем справедливы: каждый «суперпрограммист» отыскивает себе достаточно «твердый орешек», становясь, таким образом, «смиренным программистом»; и, наверное, у каждого программиста непременно находится свой круг задач, решая которые, он чувствует себя «суперпрограммистом».

Применяя ЭВМ в качестве объективного критерия при определении успешности своей работы, программист резко ужесточает допуски на точность результатов: нельзя сказать, что «текст программы достаточно хорош, так как в нем только одна тысячная процента символов ошибочны». Этот эффект еще более обусловлен системным характером любого процесса понимания. *Программист исправляет ошибки лишь сознательным усилием*, поэтому ему невозможно сохранять иллюзию своей непогрешимости.

Объективный критерий в оценке работы программиста помогает различить профессиональные и социальные факторы, влияющие на судьбу исследовательских и технологических разработок.

Важная *особенность работы программиста — ее неустранимо коллективный характер.* Даже самый замкнутый программист-одиночка работает «в коллективе» с собой-вчера и собой-завтра, с собой-до отпуска и собой-после гриппа, с собой-проектировщиком и собой-внедренцем, с собой-заказчиком и собой-пользователем, не говоря уже об использовании инструментальных систем, методов, принципов, решений, результатов других людей — информации об успехах, затруднениях и неудачах в чужих разработках.

При этом *вся конкретная работа (и познавательная, и техническая) осуществляется пока сугубо индивидуально* — неожиданная передача незавершенной работы часто аннулирует почти все промежуточные результаты.

Неудобно, что *у программиста нет и не может быть такого лидера, как конструктор или технолог, который бы заранее знал, какой конкретно должна быть готовая программа.* Заказчики, координаторы, консультанты и эксперты влияют лишь на отдельные аспекты представления программы.

Резюмировать можно тезисом: **программирование — это индивидуальное осуществление коллективной деятельности по накоплению все более точных знаний в форме программ, выполнимых на ЭВМ.**

Сделаем практические выводы относительно гигиены труда программиста. *Большинство затруднений в работе программиста обусловлены характером его труда — необходимостью индивидуально перерабатывать большие потоки информации при ограниченной пропускной способности самого программиста.* Возникающие специфические формы усталости бесполезно игнорировать или преодолевать волевыми усилиями. Для снятия такой усталости необходимо выйти из ситуации: зайдя в тупик, полезно переключиться на «фоновую работу», организовать себе активный отдых, или полноценный сон.

О РАБОТЕ ПРОГРАММИСТА

В программировании, как, вероятно, и всюду, профессионализм проявляется в умении преодолеть («обвести вокруг пальца») неподатливость объекта приложения сил, сложность коллективной деятельности и свою собственную неорганизованность.

Борясь с этими «врагами», программист постоянно ощущает ограниченность своих сил. Обычно ему удается компенсировать эту ограниченность рационализацией стиля работы. К сожалению, пока что у программистов нет возможности воспринимать правильный стиль в процессе обучения ремеслу. Каждый вырабатывает свой стиль, уже в ходе профессиональной деятельности пополняя свой арсенал удачных приемов работы и осмысливая их в виде некоторых основных принципов. Рассмотрим, без претензий на авторство, некоторые из них:

Архитектура программы должна быть «естественной» и должна обусловливаться структурой предметной области задачи.

Очень часто при изучении сложных на первый взгляд задач удается обнаружить их глубинную простоту. Поэтому, систематически выявляя естественную структуру задачи, можно снизить нагрузку на память программиста, а на этапе сопровожде-

ния — облегчить изучение и целесообразное применение итоговых решений, ввиду простоты представления.

Программирование должно быть результативным — следует на возможно более раннем этапе доводить работу до состояния «продукта», пригодного для использования. Это позволяет своевременно увидеть и оценить результаты своего труда и эффективно применить их для организации дальнейшей работы.

Результат работы, вообще говоря, не обязательно является действующей программой. Сама возможность создания действующей программы заранее не очевидна. Она исследуется при проектировании программы и обеспечивается тщательно подобранными стратегией и тактикой изучения предметной области при решении задачи. Поэтому даже в тех случаях, когда, скажем, из-за нехватки доступных ресурсов, получить программу-продукт не удается, полезный результат — это повышение уровня изученности решаемой задачи.

Кропотливая работа должна выполняться только один раз.

Одно из достижений вычислительной техники — технология абсолютно точного тиражирования. Она позволяет получить копию, полностью идентичную оригиналу (если строго, то отличить точную копию от неточной и сделать отбраковку). Это, в свою очередь, позволяет сделать сходящимся процесс исправления ошибок.

Например, использование современных средств вычислительной техники позволяет «выловить» все опечатки в большом тексте, так как при каждом сеансе редактирования объем изменяемого текста становится все меньше, а остальной текст просто копируется без внесения новых ошибок.

Именно эта технология дает нетривиальную возможность ненадежными руками на ненадежном оборудовании создавать правильные тексты программ длиной в тысячи строк без единой опечатки — а одна-единственная опечатка полностью разрушает работу программы! (Уточним, что под правильностью в данном контексте понимается лишь отсутствие технических ошибок — *синтаксических* ошибок и опечаток. Функциональная — *семантическая* — правильность программ — это нечто другое и здесь не рассматривается). В результате мы можем со стопроцентной уверенностью судить о правильности функционирования программы даже на ненадежно работающей машине.

Эту методику создания и поддержания безошибочных текстов программ можно охарактеризовать как «накопление правильности». Она позволяет программисту систематически освобождать себя от чисто технической работы по «ловле блох» и с гарантией делать надежно работающие системы. К слову сказать, перегруженность технической работой зачастую свидетельствует лишь о небрежном отношении к организации своего труда, о неприязни к смене навыков, или о стремлении произвести впечатление «делового человека».

К сожалению, еще не выработана техника программирования, которая автоматически обеспечивала бы соблюдение сформулированных принципов. Однако (еще не умеем, но уже хотим) эти принципы хорошо просматриваются в требованиях, которые программирование предъявляет к форме результатов.

Так, например, «свертка целостно используемых фрагментов» и «синтаксическая отделимость связанных по смыслу частей» способствуют выявлению естественной архитектуры, так как позволяют скрыть случайные взаимосвязи и акцентировать внимание на характерных особенностях проблемы.

«Обеспечение обратной связи с пользователем» также помогает выявить естественную архитектуру, способствуя, кроме того, некоторому разделению труда по подготовке текстового материала и выполнению отладки.

Далее «минимизация целостно программируемых компонент» хорошо коррелирует с требованием результативности, поскольку для малых программ не проблема — получить результат за приемлемое время.

«Выделение типовых и базисных модулей» обеспечивает накопление правильности и освобождает от повторного выполнения сделанной работы, одновременно гарантируя надежность программ, использующих такие модули.

«Разгрузка последующих этапов работы», обеспечивая результативность программирования, способствует своевременному прогнозированию возможных сюрпризов и, следовательно, профилактике фатальных ситуаций.

Разумеется, правильное функционирование программы немыслимо без доказательности всех построений и преобразований, выполняемых на протяжении не только фазы собственно построения текста программы, но и всего жизненного цикла программы.

В заключение отметим, что приятной особенностью программирования является отсутствие принципиальных ограничений на овеществление (в форме программ) успешного практического опыта. Требуется лишь достаточно оперативно осознать значимость своих находок и уметь выделять эти находки в чистом виде.

ГИБКОЕ ПРОМЫШЛЕННОЕ ПРЕДПРИЯТИЕ

Сокращение ГПС — гибкая производственная система — стало уже привычным. Теперь появилось еще одно: ГПП — гибкое промышленное предприятие.

Первое ГПП ввела в строй японская фирма «Ямадзаки» — один из крупнейших в мире производителей систем ЧПУ (400 штук в месяц). Завод в Нагое занимает 2790 м², стоит 18 млн. долл. Ожидается, что окупится он за 2,5 года. Значительная доля трудозатрат при создании завода пришлась на программирование — 25 тысяч человеко-часов, не считая программирования обработки деталей на обрабатывающих центрах. Всего таких центров 18. Они объединены в короткую (90 м) и длинную (108 м) линии, на каждой установлен транспортный робот.

На предприятии в две смены занято по 6 человек, а с полуночи до 8 часов утра на нем вовсе нет работников, одни роботы. ГПП выпускает 74 наименования изделий весом от нескольких килограммов до 8 т, за месяц производится до 1400 деталей. Характерно, что все детали изготавливаются из одного и того же легкообрабатываемого сплава.

Центральная ЭВМ предприятия регулярно выдает подробную документацию о ходе технологического процесса. Устройства вывода на печать информации работают круглосуточно, едва хватает времени, чтобы оформить необходимые сопроводительные документы.

«ЭКО». «Moderní Řizeni» (ЧССР), 1983, № 6.

«Скорая» для ЭВМ

ТАШКЕНТ (ТАСС). Причину выхода из строя микропроцессорной техники способен быстро находить анализатор, созданный учеными объединения «Кибернетика» Академии наук Узбекистана. Обнаружив неполадку, новое устройство выдает на экран дисплея изображение поврежденного участка схемы. Это облегчает ремонт электронной техники.

Новинку можно использовать также при изготовлении микропроцессорных комплексов, компьютеров, диалоговых и цифровых устройств. Она строго следит за технологией сборки, проверяет качество.

«Правда», 9 августа 1984 г.

УДК 681.325.5—181.48.004:621.39

Г. Г. Кудрявцев, И. А. Мамзелев

МИКРОПРОЦЕССОРНЫЕ СРЕДСТВА В ТЕХНИКЕ СВЯЗИ

(обзор по отечественным данным)

Широкое применение микропроцессорных средств в отрасли связи от сложных систем коммутации и передачи информации до телефонного аппарата позволяет повысить эффективность использования оборудования и создает основу для внедрения перспективных технологий обслуживания абонентов.

В настоящее время связь превратилась в одну из ответственных отраслей народного хозяйства, темпы развития которой значительно опережают темпы развития многих других отраслей [1]. Каждый рубль капитальных затрат, вложенный на развитие электросвязи дает 2,5—3 руб. экономии затрат в материальном производстве, либо 3,5 руб. прироста валового общественного продукта и 1,7 руб. прироста национального дохода [2].

Развитие отрасли связи неразрывно связано с ускорением научно-технического прогресса, внедрением новых технологий обслуживания абонентов на основе применения современных средств микроэлектроники и вычислительной техники, обеспечивающих значительное повышение уровня автоматизации основных технологических процессов и производительности труда в отрасли [3]. Широкому внедрению вычислительной техники в отрасль связи способствует то, что сообщения, передаваемые с помощью средств электросвязи, в большинстве случаев уже представлены в форме, удобной для обработки на ЭВМ (например, телеграфные сообщения) или легко могут быть представлены в такой форме, поскольку переносчиком сообщения является аналоговый либо цифровой электрический сигнал.

Интенсивное использование вычислительной техники в отрасли стало возможным лишь с появлением массовых микропроцессорных средств вычислительной техники (МСВТ с широкими функциональными возможностями) [4]. Каждая подотрасль отрасли связи (телефония, телеграфия) оснащается своей типовой аппаратурой, уровень межотраслевой унификации которой в настоящее время весьма невысок, что существенно усложняет процессы эксплуатации, ремонта, подготовки кадров и т. д. Появление МСВТ создает предпосылки для создания унифицированных технических средств.

Таким образом, одним из ведущих направлений технического прогресса в отрасли связи является комплексная автоматизация технологических процессов и процессов управления на основе применения МСВТ, направленная на сокращение затрат ручного труда с одновременным расширением функций и услуг связи.

Основные функции отрасли связи. Требования, предъявляемые к МСВТ. Основные целевые функции отрасли связи состоят в предоставлении потребителям технических возможностей для обмена информацией в виде сообщений и почтовых отправлений, дополнительные — в обеспечении заданного качества обмена, учета и оплаты предоставленных услуг. К техническим средствам отрасли, реализующим основные функции, относятся сети электрической и почтовой связи. Дополнительные функции выполняются с помощью автоматизированных систем управления и технического обслуживания средств связи. В свою очередь каждую из рассмотренных основных функций можно рассматривать как совокупность различных функций.

Например, процесс обмена информацией складывается из преобразования сообщения от источника информации в электрический сигнал и передачи его в сеть электросвязи, распределения (коммутации сообщений) между источником и потребителем, приема сообщений, обратного преобразования и выдачи сообщения потребителю.

Выбор соответствующих конструктивных модулей унифицированных технических средств для всех подотраслей осуществляется на основе обобщенных функций, к которым можно отнести преобразование сообщения в электрический сигнал, передачу, прием, распределение, учет, обслуживание клиентуры и абонентов, техническое обслуживание средств связи. Анализ обобщенных функций свидетельствует о том, что при их реализации в большинстве случаев требуется сложная логическая обработка информации, поэтому в качестве типового унифицированного модуля целесообразно использовать законченную одноплатную или однокристальную микроЭВМ. Проблемная ориентация модуля должна осуществляться с помощью набора микроконтроллеров и лишь в отдельных случаях, при отсутствии микроЭВМ с соответствующими показателями, целесообразна реализация отдельных функций на основе МП систем или матричных БИС.

Учитывая функции аппаратуры связи, сформулируем общие требования, которым должны отвечать МСВТ, применяемые в отрасли. Стоимость единицы вычислительного ресурса, иначе удельная стоимость операции в секунду, должна составлять 0,01—0,001 руб при объеме памяти 10^6—10^7 бит и производительности 10^5—10^6 операций/с. Для решения большей части задач должно использоваться ограниченное число типов микропроцессоров и микроЭВМ. Вычислительные средства должны обеспечивать реализацию параллельных процессов (обслуживание групп абонентов, потоков заявок и т. д.) наиболее простым способом; обладать программной совместимостью и способностью к постоянному наращиванию при увеличении числа объектов и расширении круга задач.

Повышенное требование отрасль предъявляет к живучести систем (возможности функционирования при отказах элементов) и продолжительности срока служ-

бы (системы и оборудование связи должны иметь срок службы ~20 лет). Большинство автоматизированных систем отрасли является территориально-распределенным, поэтому одно из основных требований к вычислительным средствам состоит в возможности реализации принципа распределенной обработки информации, что позволит существенно снизить требования к пропускной способности каналов связи.

Рассмотренным требованиям наиболее полно отвечают вычислительные средства, использующие МСВТ и построенные на основе концепций коллектива вычислителей [5], управляющего коллектива вычислителей [6] и модульных асинхронных систем [7]. Концепция управляющего коллектива обеспечивает параллельность выполнения операций, программируемость логической структуры, модульность конструктивного исполнения, возможность распределенной обработки информации, отказоустойчивое функционирование как вычислительного ядра, так и средств связи с объектами управления и обслуживания. Подобные системы и являются унифицированными техническими средствами.

При таком подходе для реализации большинства функций используется базовая вычислительная система [8], состоящая из совокупности элементарных машин (ЭМ), объединенных сетью связи и сопрягаемой с объектами контроля и управления с помощью микропроцессорных устройств связи с объектами. ЭМ состоит из вычислительного модуля, в качестве которого используются серийные микроЭВМ «Электроника 60», «Электроника НЦ-80-01Д» и модуля системного интерфейса, обеспечивающего функционирование набора ЭМ как единой системы.

Система снабжается типовым набором средств хранения и отображения информации и, в зависимости от области применения, интерфейсными платами, реализованными на основе МП систем, обеспечивающими отказоустойчивое сопряжение вычислительной системы с каналами связи и объектами управления и обслуживания.

Локальные вычислительные системы, обслуживающие объекты отрасли, объединяются в распределенные вычислительные системы района, города, области и т. д., образуя многоуровневые системы с распределенной обработкой информации и децентрализованным управлением. Подобная концепция позволяет использовать микропроцессорные средства с ограниченными ресурсами для создания вычислительных и управляющих систем с широким спектром функций и большими возможностями.

Микропроцессорные средства в аппаратуре обмена и распределения информации. В настоящее время 90—98% существующих каналов связи используется для телефонии. Согласно прогнозу развития связи США общий среднегодовой объем телефонных сообщений составит на 1990 год 10^{19} бит (в предположении, что будут использованы телефонные каналы с ИКМ со скоростью 64 Кбит/с), среднегодовой объем сообщений во всех остальных видах электросвязи должен составить 10^{17} бит [9].

Наиболее массовое применение МСВТ найдут в оконечных устройствах телефонных сетей — телефонных аппаратах, которых в настоящее время насчитывается более $25 \cdot 10^6$ штук. МП БИС встраиваемые в телефонный аппарат аналогового типа, значительно расширяют возможности абонентов, обеспечивают возможность тестатурного набора номера, повторный набор последнего номера, сокращенный набор для наиболее часто вызываемых абонентов и т. д.

Использование МП БИС позволяет перейти к принципиально новым системам расчетов с абонентами за междугородные разговоры, что позволит существенно сократить персонал, занятый этим видом работы. Телефонный аппарат на основе МП обеспечивает определение набираемого номера, тарифа и стоимости разговора. Оплата осуществляется при помощи специальной платежной карты, например, магнитного типа, приобретаемой абонентом заранее и помещаемой на время разговора в аппарат. Карта рассчитана на определенную длительность разговора. По мере использования платежная способность карты уменьшается путем стирания хранящейся в ней информации. Подобный аппарат может совмещать в себе функции электронных часов, индикатора вызываемого номера и использоваться при введении повременного учета.

В цифровом телефонном аппарате МП выполняет функции кодека — устройства, предназначенного для кодирования звукового сигнала и формирования потока данных со скоростью 64 Кбит/с при передаче, а также декодирования и преобразования цифрового сигнала в аналоговый при приеме. В телефонных системах с аналоговыми системами передачи и цифровой коммутацией кодек входит в оборудование АТС. Учитывая массовый характер применения кодеков целесообразно создание для этой цели специализированных БИС [10].

В цифровых системах передачи с ИКМ микропроцессорные средства занимают ведущее место в цифровых фильтрах приемников одночастных и многочастных сигналов; трансмультиплексорах, используемых при сопряжении аналоговых и цифровых систем передачи различных иерархических уровней; в эхозаградителях на междугородных линиях связи и т. д. Массовое применение при создании цифровых систем передачи найдут аналоговые МП [11], в состав которых входят 9-разрядные АЦП и ЦАП и 25-разрядный процессор. Они могут успешно использоваться в телефонных системах с многочастотной сигнализацией и при обработке речевых сигналов.

Для выполнения функций преобразования, приема и передачи, особенно в системах цифровой обработки, целесообразно применять как быстродействующие МП серии К589 и К1800, так и функционально-ориентированные БИС матричного типа, обеспечивающие повышенное быстродействие за счет отказа от микропрограммного управленяи и реализации принципов жесткой логики.

Большое потребление МСВТ и специализированных коммутационных БИС предполагается в системах распределении информации — городских и междугородных автоматических квазиэлектронных и электронных станциях [6, 7, 10]. Для создания парка в $(25—50) \cdot 10^6$ телефонов с учетом дополнительных видов услуг потребуется ввести вычислительные мощности с суммарной производительностью до $10^{10}—10^{11}$ операций/с в составе систем распределения [6]. При этом с экономической точки зрения необходима ориентация на управляющие комплексы (УК), широко использующие МСВТ. Анализ эволюции принципов управления на узлах коммутации [12] свидетельствует о том, что наиболее перспективными с точки зрения основных принципов установления телефонных соединений, являются однородные УК с программируемой структурой, реализующие принципы коллектива вычислителей и управляющего коллектива [12, 13]. Использование принципов коллектива обеспечивает естественную реализацию параллельных процессов, происходящих при обслуживании коллективов абонентов, упрощает структуру УК и программное обеспечение. УК, созданные на принципах управляющего коллектива позволяют легко выполнить любой из известных способов обслуживания вызовов (с разделением источников нагрузки, функций и нагрузки) и любой способ подключения управляющих микроЭВМ к коммутационным блокам (одна микроЭВМ на один блок, одна микроЭВМ на K блоков, n микроЭВМ на K блоков), а также обеспечивают возможность построения станций и узлов из унифицированных модулей.

Модульная структура устройства управления и станции в целом в сочетании с распределенным управлением является более экономичной, чем система с одним процессором и централизованным управлением. При организации централизованного управления, например

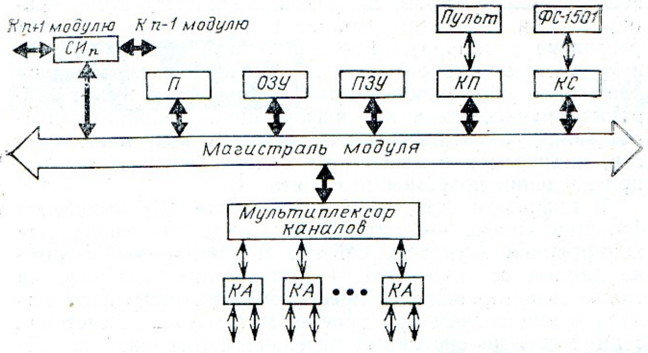


Рис. 1. Структура блока электронного телеграфного концентратора:
П — процессор К1801ВМ; КП, КС — контроллеры пульта и фотосчитывателя; КА — канальный адаптер; СИ — системный интерфейс

при помощи УК «Нева», последний должен устанавливаться сразу, в полном объеме независимо от емкости станции, и, пока станция не наберет мощность, на которую рассчитан УК, он будет недогружен. Применение станций, использующих модель управляющего коллектива, позволяет вводить их в эксплуатацию отдельными модулями по мере увеличения нагрузки, не замораживая капитальных затрат. Кроме этого, подобный подход позволяет создать на единой аппаратно-программной основе несколько станций различной емкости.

Использование принципов управляющего коллектива при создании УК дает возможность применить в качестве управляющих модулей микроЭВМ «Электроника 60». Стоимость подобных систем будет на несколько порядков ниже стоимости систем с УК «Нева».

Существенные причины, сдерживающие развитие сети телефонной связи, кроются в больших затратах труда и материалов, в частности меди, при строительстве абонентских линий, поскольку каждый абонент имеет непосредственное соединение с АТС. Сократить эти затраты можно путем установки концентраторов (или подстанций) абонентских линий с микропроцессорным управлением в местах концентрации телефонов, например, в доме или группе близкорасположенных домов. Абоненты подключаются к концентратору короткими линиями и число соединительных линий, прокладываемых к АТС, значительно сокращается. Подобные концентраторы, выполняя функции микроузла коммутации и работая в необслуживаемом режиме, должны обладать повышенной надежностью и иметь малое энергопотребление. Концентраторы емкостью до 4000 номеров [9] могут быть созданы на основе микропроцессоров серии К588, объединенных в управляющий коллектив.

Важное место займут МСВТ в сетях с дискретной информацией — телеграфных, передачи данных и т. д. Информация в таких сетях представлена в двоичнокодированном виде, и применение МСВТ дает быстрый эффект, поскольку позволяет резко повысить производительность труда и высвободить персонал при минимальных затратах, связанных лишь с согласованием уровней сигналов.

Наиболее массовое применение МП средства найдут в оконечных телеграфных устройствах, предназначенных для преобразования сообщения в электрический сигнал. В телеграфии в настоящее время существуют два типа оконечных устройств — телеграфный аппарат и аппарат факсимильной связи. Примером телеграфного аппарата нового типа является отечественный телеграфный дисплей «Элит-Т» [3], предназначенный для оснащения рабочих мест телеграфистов в оконечных пунктах общегосударственной и ведомственных телеграфных сетей.

Использование МП техники стимулирует развитие цифровой факсимальной связи, обладающей по сравнению с другими видами электросвязи такими достоинствами, как возможность высокой степени автоматизации, универсальность передаваемой информации — передача графических изображений и текста на любом языке, высокая достоверность и документальность. В настоящее время развернуты работы по созданию комплекса современной автоматизированной факсимильной аппаратуры. В состав оконечной аппаратуры этого комплекса должны войти цифровой факсимильный аппарат для автоматической передачи корреспонденции по коммутируемой телефонной сети общего пользования и сетям передачи данных, комбинированная факсимильно-телеграфная установка, обеспечивающая комплексную автоматизацию процессов обработки традиционных и факсимильных телеграмм и взаимодействие как с факсимильной, так и с символьно-кодовой аппаратурой. С вводом в действие этого комплекса появится возможность принимать телеграммы, написанные от руки. В качестве основного модуля обработки информации при создании этой аппаратуры целесообразно применять одноплатные микроЭВМ.

В существующих сетях телеграфной связи и низкоскоростной сети передачи данных ПД-200 распределение информации осуществляется при помощи коммутационных станций. До недавнего времени в телеграфных коммутационных станциях в основном использовался электромеханический — координатный способ установки соединений как наиболее дешевый. Внедрение МСВТ открывает перспективы создания электронных коммутационных станций с повышенными технико-экономическими показателями. Ведется разработка интегральных станций для коммутации низкоскоростных (50—200 бит/с) и среднескоростных (600—9600 бит/с) каналов. Для уменьшения объема работ по созданию базового вычислительного модуля и программного обеспечения в качестве основы в данных станциях используется микроЭВМ «Электроника 60» и управляющий коллектив из микроЭВМ. В узловых пунктах с большой нагрузкой применяются электронные центры коммута-

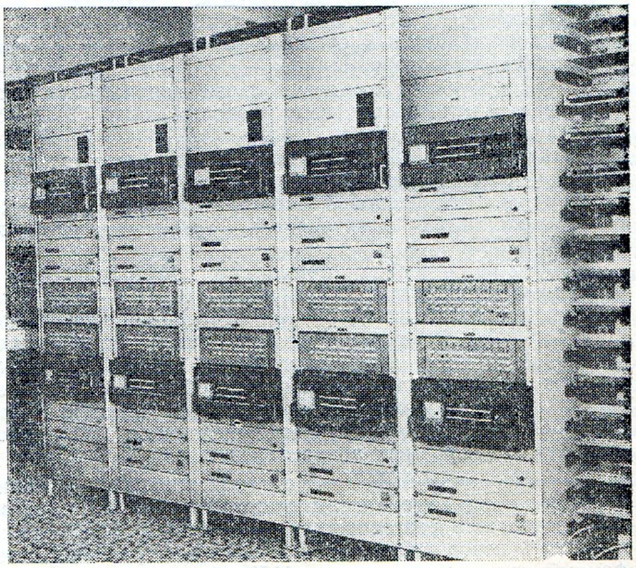


Рис. 2. Аппаратно-программный комплекс «Телеграф»

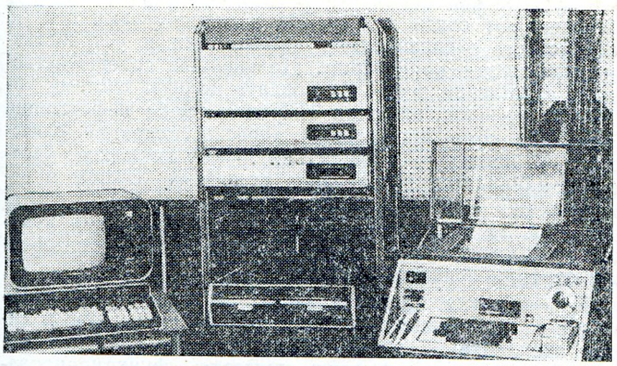


Рис. 3. Управляющий вычислительный комплекс «Связь 1»

ции сообщений телеграфного типа (ЦКС), реализующие метод коммутации сообщений. В настоящее время техническую основу ЦКС составляет вычислительный комплекс на основе двух ЭВМ типа ЕС-1030.

Наряду с применением в ЦКС, микропроцессорные средства получают широкое распространение и при создании электронных телеграфных концентраторов (ЭТК) для автоматизированного приема, обработки и передачи сообщений по некоммутируемым каналам в узлах связи. В настоящее время на базе микроЭВМ «Электроника НЦ-80-01Д» создан ЭТК, предназначенный для автоматической коммутации и обработки телеграмм в областных и крупных районных узлах телеграфной сети общего пользования (рис. 1). Схема связи ЭТК обеспечивает телеграфный обмен оконечных пунктов своей зоны с ЦКС. Максимальная конфигурация ЭТК включает 8 модулей на 128 каналов с пропускной способностью 0,7 телеграмм/с.

На основе микроЭВМ «Электроника НЦ-03С» создан и повсеместно внедряется на сети прямых соединений аппаратно-программный комплекс «Телеграф» [14], позволяющий автоматизировать процесс переприема телеграмм в узлах (рис. 2). Комплекс имеет модульную структуру, что обеспечивает простое наращивание модулей в зависимости от нагрузки и числа точек подключения. В состав комплекса входят, кроме микроЭВМ с инженерным пультом, блоки канальных адаптеров (телеграфных контроллеров) и устройства преобразования сигналов. Канальный адаптер обслуживает одновременно четыре канала и обеспечивает позначный обмен информацией. Общая производительность МСВТ, требуемых для передачи данных с учетом абонентских терминалов, может быть оценена в 10^8—10^9 операций/с.

В перспективе внедрение вычислительной техники и цифровых систем передачи обеспечит построение интегральной цифровой сети связи (ИЦСС), в которой для коммутации каналов, пакетов и сообщений будут применяться общие узлы коммутации [6]. Поэтому, рассматривая применение МСВТ в аппаратуре, предназначенной для существующих сетей электросвязи, следует выбирать такие технические решения, которые позволят в дальнейшем успешно перейти к ИЦСС.

Степень автоматизации технологических процессов связи, реализующих основные функции, весьма высока. Большинство этих процессов выполняется автоматически без вмешательства оператора, поэтому основная цель внедрения микропроцессорной техники в подобную аппаратуру состоит в снижении ее стоимости, уменьшении потребляемой мощности, повышении надежности функционирования, упрощении технического обслуживания, повышении степени унификации, расширении функциональных и технических возможностей, уменьшении массогабаритных характеристик, увеличении числа обслуживаемых абонентов при сохранении тех же площадей. В конечном итоге все это способствует росту числа абонентов на единицу обслуживающего персонала, т. е. повышению производительности труда в отрасли.

Микропроцессорные средства в почтовой связи и автоматизированных системах технической эксплуатации. Анализ трудозатрат в отрасли показал, что 91,1% работающих в отрасли составляют работники девяти основных профессий: почтальоны — 23,67%, инженерно-технические работники, эксплуатирующие аппаратуру, реализующую дополнительные функции, электромонтеры эксплуатационники — 13,64%, операторы — 8,52%, телефонисты — 6,96%, телеграфисты — 5,67%, водители — 4,4%, сортировщики — 3,49%, киоскеры — 2,8%. Для количественной оценки масштабов внедрения МСВТ, например, в почтовой связи, отметим, что в настоящее время предприятиями почтовой связи, число которых близко к $9 \cdot 10^4$, ежегодно обрабатывается и доставляется 52 млрд. различных отправлений весом 20 млн. тонн, ежедневно доставляется примерно 380 млн. газет и журналов, проходит в год 150 млрд. рублей денежного оборота и при этом большое число операций по обработке почтовых отправлений выполняется вручную.

Выше было показано, что требованиям отрасли при реализации обобщенных функций наиболее полно отвечает управляющий коллектив, который и используется в настоящее время при создании комплексов технических средств автоматизированных систем управления и технической эксплуатации. Применение в качестве вычислительного модуля серийных микроЭВМ позволило достаточно быстро получить практические результаты в создании подобных систем. Поскольку проблемы применения модели управляющего коллектива на основе МСВТ в системах технической эксплуатации отрасли достаточно подробно рассмотрены в работе [15], перечислим лишь наиболее важные из разрабатываемых систем и рассмотрим уже созданные системы, которые условно можно разделить на три группы.

К *первой группе* относятся автоматизированные системы управления, контроля и обслуживания средств связи. В их состав входят системы оперативно-технического управления и контроля магистральной первичной

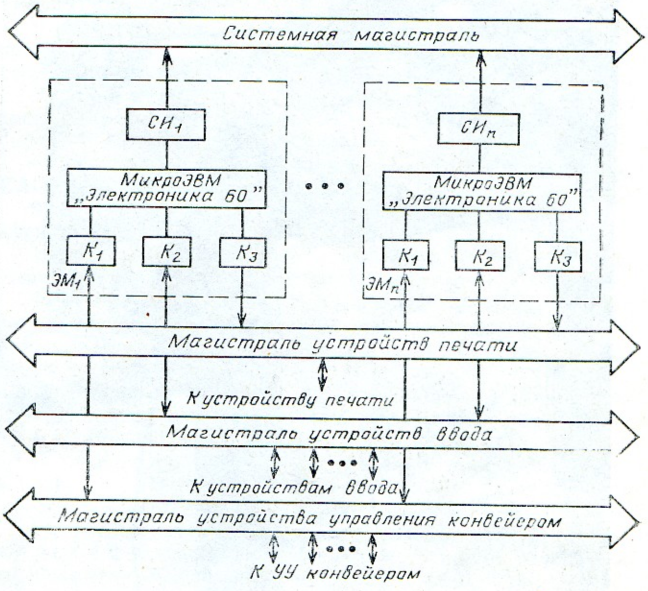


Рис. 4. Структура управляющего устройства комплекса сортировки тяжелых почтовых отправлений: СИ — системный интерфейс

сети связи, технической эксплуатации городских и сельских телефонных станций, управления телеграфными сетями общего пользования, абонентского телеграфа и передачи данных, управления сортировкой газетно-журнальной экспедиции Союзпечати и т. д.

Техническими средствами, обеспечивающими работу указанных систем, являются унифицированные информационно-управляющие комплексы, устанавливаемые в местах возникновения информации — линейных аппаратных цехах, в зданиях АТС, местах сортировки и т. д. Примером структуры системы подобного типа может служить модульный управляющий вычислительный комплекс «Связь 1» [6, 15], прдназначенный для контроля состояния оборудования (рис. 3). Вычислительное ядро комплекса образовано объединением трех ЭМ на основе микроЭВМ «Электроника 60» в кольцевую макроструктуру. Отказоустойчивое сопряжение с объектами контроля осуществляется через интерфейс, в состав которого входят платы И2 и контроллеры связи с объектами. В системе использованы датчики типа «сухой контакт». Система контроля оснащена необходимыми периферийными и запоминающими устройствами. При необходимости отдельные триады объединяются в кольцевую структуру через устройства последовательного обмена.

Более высокие надежностные характеристики при объединении достаточно большого числа ЭМ имеет управляющий коллектив с магистральной организацией. Одна из таких систем входит в комплекс сортировки тяжелых почтовых отправлений (посылок, бандеролей) (рис. 4) и позволяет распознавать информацию о посылке по нанесенному на ней семизначному десятичному числу, вырабатывать сигналы управления сортировочным конвейером, направляющим посылку к заданному накопителю, и формировать сопроводительную документацию. При отказах ЭМ система автоматически изменяет конфигурацию и продолжает функционирование. Межмагистральные контроллеры $К_1$—$К_3$ совместно с магистралями и адаптерами внешних устройств обеспечивают связь между ЭМ и устройствами печати, устройством распознавания информации и устройствами управления конвейером. Дальнейшее совершенствование комплекса позволит создать полную систему обработки посылок, включая операции ввода информации с бланков сопроводительных адресов, операции сличения, т. е. проверки соответствия между посылкой, адресом и накладной. Применение подобных систем в несколько раз повышает производительность труда сортировщиков за счет отказа от ручного оформления документации, а также повышает коэффициент использования сортировочных установок.

Дешевизна и простота МСВТ позволяет применять их на самых нижних уровнях автоматизированных систем эксплуатации аппаратуры связи, что дает возможность существенно повысить качество функционирования электромеханической аппаратуры распределения информации благодаря введению статистических контрольно-корректирующих методов эксплуатации. Вмешательство персонала для устранения неисправностей необходимо только тогда, когда потери достигают заданного уровня. Важность внедрения этих методов становится более понятной если учесть, что в настоящее время электромеханическая аппаратура распределения информации обслуживает более $20 \cdot 10^6$ абонентов. Внедрение МСВТ дает возможность на основе статистического контроля применять прогнозные методы эксплуатации, что кроме освобождения персонала существенно повышает надежность связи.

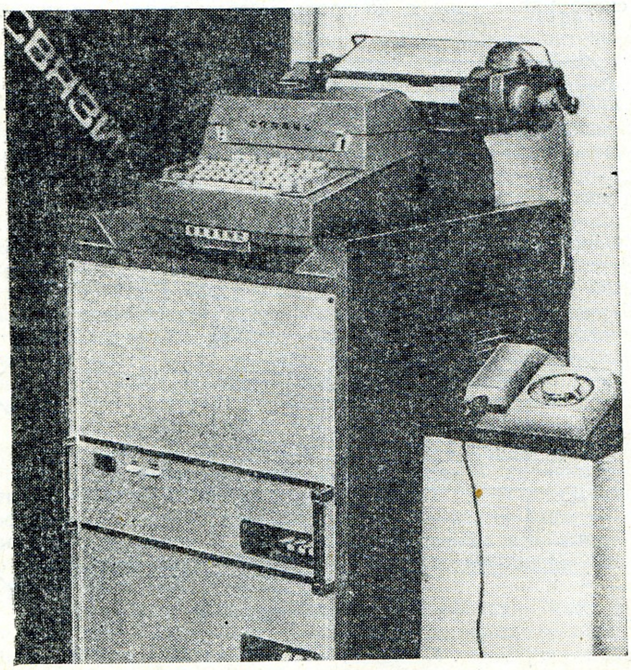


Рис. 5. Система контроля качества соединений для телефонной сети

На рис. 5 показана система контроля соединений для телефонной сети, обеспечивающая слежение за обслуживанием от момента начала вызова до появления разговорных сигналов или до момента определения результата вызова, измерение и регистрацию продолжительности этапов установления соединений, регистрацию набираемых номеров, определение результатов попыток вхождения в связь и качества разговорных трактов, группирование соединений по результатам вхождения в связь по зонам, районам и станциям для расчета коэффициентов тяготения и полезной работы сети, передачу информации в центр технического обслуживания. Ядром системы служат комплексы «Связь-1», подключаемые через интерфейсные платы и блоки концентрации ко входам АТС. Приборы для автоматического измерения параметров аппаратуры и трактов в автоматизированных системах технической эксплуатации на базе МСВТ разрабатываются на основе МП серий К580 и К589.

Вторую группу составляют автоматизированные системы по расчетам с абонентами и клиентурой: системы заказов и расчетов за междугородные телефоны, повременной оплаты за городской телефон, комплексных расчетов за услуги электросвязи и т. д. К ним относится, например, аппаратура «Отель», предназначенная для расчетов с предприятиями и учреждениями за междугородные телефонные разговоры, а в перспективе и за местные разговоры с повременной системой учета (рис. 6). В состав модуля аппаратуры входит микроЭВМ «Электроника 60», дополненная периферийным процессором и модулями согласующих устройств для подключения к абонентским линиям. Для повышения надежности два модуля могут обслуживать одни и те же линии либо объединяться в управляющий коллектив. Аппаратура «Отель» обеспечивает определение состояния абонентской линии телефонной сети, регистрацию данных о телефонных разговорах, тарификацию телефонных разговоров, документирование операций по расчету с клиентурой за услуги связи, контроль и оповещение о выходе телефонов из строя и т. д. Один модуль обеспечивает подключение до 1000 контролируемых телефонов при 128 одновременных телефонных разговорах.

В *третью группу* входят автоматизированные системы управления районными и выделенными предприяти-

ями связи, система сбора и обработки информации для предприятий и отделений связи и т. д. Техническими средствами, обеспечивающими функционирование систем последних двух групп, являются многофункциональный терминал предприятия связи и центральное оборудование на базе МСВТ.

Для построения автоматизированных систем могут быть использованы микроЭВМ «Электроника 60», «Электроника-НЦ-80-01Д», диалоговые вычислительные комплексы «Электроника-НЦ-80-20». При разработке системных интерфейсов целесообразно применять быстродействующие МП БИС серии К589, К1800 и коммутаторные БИС или МП, совместные с МСВТ вычислительного ядра, например, серии К1801. Для построения контроллеров сопряжения со средствами связи целесообразно применять МП комплект серии К580. Широкое распространение для этих целей могут найти однокристальные микроЭВМ, аналогичные *i*8041, *i*8741 либо TMS-1000 и т. д.

Реализация рассматренных технических средств подготовит переход к следующему этапу развития техники связи, который будет характеризоваться интеграцией МСВТ, реализующих распределенную обработку со средствами связи, появлением новых видов услуг связи и новых технологий обслуживания абонентов и клиентуры, таких как электронная почта.

Основу технических средств составит интегральная цифровая сеть связи, входящая в состав ЕАСС и позволяющая объединить отдельные виды сетей в унифицированный набор средств вычислительной техники, созданный на основе следующих поколений МСВТ для техники связи. В состав средств вычислительной техники должны входить центр коммутации цифровой информации, унифицированный терминал предприятия связи, обеспечивающий выполнение всех технологических процессов, свойственных предприятию связи, унифицированный терминал учрежденческого типа и терминал абонента (рис. 7), гарантирующий каждому абоненту возможности телефонной, телеграфной, документальной видов связи, а также справочно-информационное обслуживание [3]. Унифицированные терминалы будут оснащены синтезаторами и анализаторами речи, дистанционными читающими и пишущими автоматами, обеспечивающими общение с человеко-машинными справочно-информационными комплексами. Особенность этого этапа внедрения средств вычислительной техники заключается также и в том, что общие технические средства будут использованы одновременно для решения всех классов задач, т. е. произойдет интеграция решаемых задач.

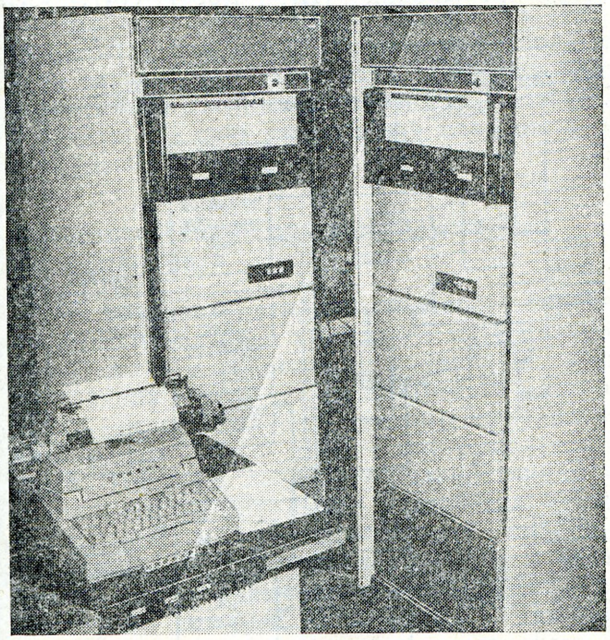


Рис. 6. Система учета стоимости междугородных телефонных разговоров «Отель»

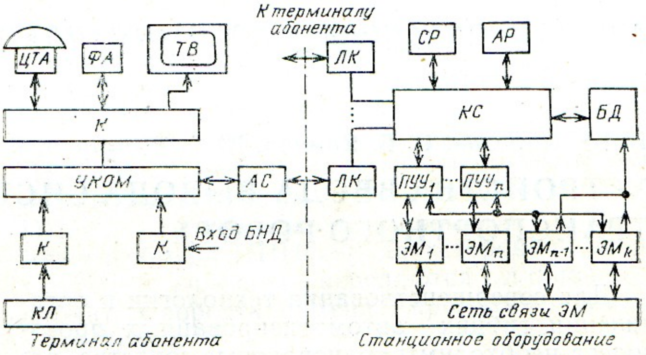


Рис. 7. Обобщенная структура перспективной системы телеинформационного обслуживания

ЛИТЕРАТУРА

1. Шамшин В. А. Новый этап в планировании развития электросвязи страны. — Электросвязь, 1980, № 1.

2. Кудрявцев Г. Г. Отрасль связи и вычислительная техника. Проблемы и перспективы. XXXVIII Всесоюзная научная сессия, посвященная Дню радио. Тезисы докладов пленарных заседаний. — М.: Радио и связь, 1983.

3. Бухвинер В. Е. Телеобслуживание и человеко-машинная связь. — М.: Радио и связь, 1983. — 168 с.

4. Евреинов Э. В. Единая распределенная вычислительная система. Проблемы и перспективы развития. Вычислительные системы. — М.: Статистика, 1980, вып. 1, с. 8—25.

5. Евреинов Э. В., Хорошевский В. Н. Однородные вычислительные системы. — Новосибирск: Наука, 1978. — 320 с.

6. Мамзелев И. А. Проблемы применения микропроцессорных средств в технике связи. — Электронная промышленность, 1983, № 9, с. 31—36.

7. Марчук Г. И., Котов В. Е. Модульная асинхронная развиваемая система (концепция). — Новосибирск: ВЦ СО АН СССР, препринты, № 86, 87

8. Юзжалин А. С. Некоторые вопросы развития ЕАСС. — Электросвязь, 1981, № 4, с. 1—4.

9. Коммутация в сетях связи. — Тематический выпуск ТИИЭР, 1977, № 9. — 245 с.

10. Применение микропроцессоров. — Тематический выпуск ТИИЭР, 1978, № 2. — 198 с.

11. Горохов В. Л., Ефимов И. Е. Микроэлектроника и микропроцессоры в технике связи и телеинформационном обслуживании. — Электросвязь, 1983, № 11, с. 1—7.

12. Пшеничников А. П. Эволюция принципов управления на узлах коммутации. — Электросвязь, 1981, № 4, с. 40—44.

13. Евреинов Э. В., Ефимов И. Е., Мамзелев И. А. Принципы построения вычислительных систем в отрасли связи. — Электросвязь, 1981, № 4.

14. Калинин Б. С., Рубцов В. М., Сандомирский А. Е. Автоматизация переприема телеграмм на коммутируемых сетях. — Электросвязь, 1982, № 10.

15. Кудрявцев Г. Г., Мамзелев И. А., Часовников Е. Д. Принципы построения автоматизированных систем технической эксплуатации в отрасли связи. — Электросвязь, 1983, № 12, с. 1—11.

Статья поступила 19 марта 1984 г.

УДК 007.52:658.286

Б. Ю. Порозов, С. Б. Коннов, Ю. В. Комаров, В. Е. Черниенко

УСТРОЙСТВА ВВОДА-ВЫВОДА СИСТЕМЫ УПРАВЛЕНИЯ ТРАНСПОРТНОГО РОБОТА

Для совершенствования технологии и организации гибких автоматизированных производств необходимы транспортные средства, работающие без непосредственного участия человека. К ним относятся мобильные транспортные роботы, например автоматические тележки, при движении которых используется ведущая линия. Такие тележки не решают всего комплекса задач автоматизации напольного транспорта. Поэтому интенсивно разрабатываются интегральные транспортные роботы [1].

Одна из наиболее сложных проблем в реализации системы управления транспортным роботом — создание бортового вычислительного комплекса (БВК): алгоритмического, программного и аппаратного обеспечения с повышенной живучестью при сбоях и отказах [2, 3].

Стандартные устройства ввода-вывода для БВК отсутствуют. Это вызвало необходимость разработки многопортового интерфейса, последовательного программируемого интерфейса связи, пульта диалога оператора с вычислительным комплексом системы управления робота и контроллера кассетного накопителя.

Многопортовый интерфейс

Сопряжение с вычислительной управляющей системой значительного числа портов ввода-вывода (ПВВ) с различными режимами обмена — одна из проблем, возникающих при разработке систем управления роботами. Опыт проектирования показывает, что основную часть подсистемы ввода-вывода составляют устройства пассивного обмена и обмена с прерыванием программы. Внепроцессорный канал требуется значительно реже. Без специализированной подсистемы обмена с ПВВ в архитектуре микроЭВМ «Электроника 60» и аналогичных ей гибкость программирования выше, но аппаратура портов усложняется. В итоге на 1 байт ПВВ с прерыванием, при раздельной реализации, затрачивается примерно 8—10 корпусов интегральных схем только для соединений с каналом микроЭВМ. При ПВВ в десятки байт (для сложных систем управления роботов) такой расход недопустим. Альтернатива — групповая реализация ПВВ за счет унификации служебного обмена ПВВ с каналом микроЭВМ. Это позволяет выделить служебную часть ПВВ в отдельное устройство.

Разработанный *многопортовый интерфейс* (МИ) организует служебные операции по передаче данных из ПВВ в канал или обратно (под управлением процессора), процедуры векторного прерывания (аппаратно) и обеспечивает возможность групповой обработки запросов на ввод-вывод от ПВВ (под управлением процессора).

Структура МИ (рис. 1) состоит из трактов данных, адреса, прерываний и узла синхронизации.

Тракт данных — двунаправленный 16-разрядный переключатель, подключающий на время обмена магистраль ДА канала «Электроника 60» к магистрали ПВВ в необходимом направлении. К магистрали ПВВ подключены также служебные ПВВ самого МИ.

Тракт адреса образован селектором группы адресов (СГА), регистром адреса порта (РАП), селектором служебных портов (ССП) и согласующим усилителем адреса порта.

Тракт прерываний включает в себя процессор прерываний (БИС К589ИК14), узел

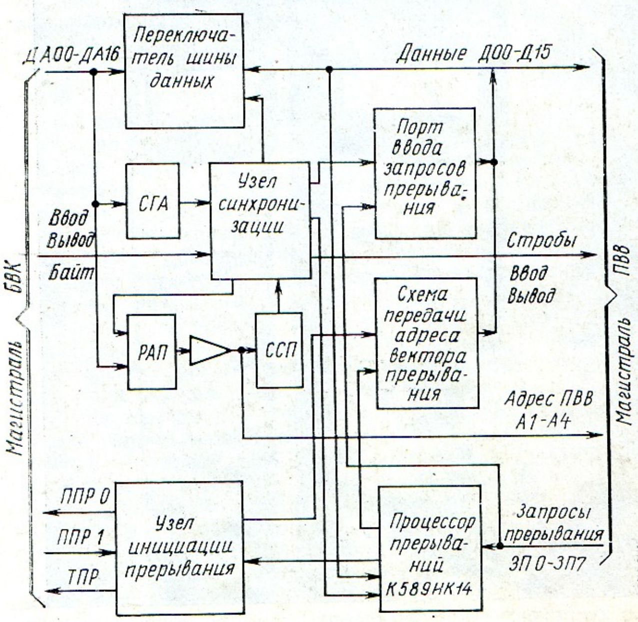


Рис. 1. Структура многопортового интерфейса

инициации прерываний и схемы передачи адреса вектора. Для групповой обработки выделен служебный байтовый порт ввода, через который программно доступны запросы прерываний на входах процессора прерываний.

Узел синхронизации вырабатывает необходимые стробы сопровождения данных на магистрали ПВВ для внешних и служебных портов и управления трактами данных и прерываний.

Версия МИ1-1, выполненная в виде платы размером 135×240 мм, устанавливается в любое свободное гнездо канала микроЭВМ «Электроника 60». МИ1-1 обеспечивает доступ к 16 или 32 байтовым портам (в зависимости от настройки) по программному каналу, один из которых служебный для тракта прерываний, и способна обрабатывать восемь запросов на прерывание с генерацией различных адресов векторов прерываний. Магистраль ПВВ выводится с МИ1-1 через дополнительный разъем на дополнительный каркас с платами, содержащими нестандартные части интерфейсов терминалов и по две интегральные схемы для подключения каждого байтового порта к магистрали ПВВ. Напряжение питания МИ1-1 +5 В, ток потребления 0,8 А.

При организации обмена с большим числом ПВВ можно использовать соответствующее число МИ1-1, поскольку старшие разряды адреса группы портов и адреса вектора прерывания программируются перемычками. Один МИ1-1 эквивалентен двум единицам нагрузки канала «Электроника 60».

Простота подключения новых портов к микроЭВМ при такой организации ввода-вывода значительно облегчает проектирование интерфейсов и снижает стоимость разработки и изделий за счет уменьшения сложности интерфейсов.

Последовательный программируемый интерфейс связи

На различных стадиях готовности бортового вычислительного комплекса интегрального транспортного робота при разработке, комплектовании и отладке аппаратного и программного обеспечения его системы управления необходима связь БВК с инструментальной ЭВМ. Эта связь позволяет разработчикам использовать терминальные устройства инструментальной машины и облегчает процедуру ввода-вывода. Для наиболее простой и эффективной связи применяется последовательный интерфейс, который используется также для связи блоков собственного транспортного робота: в частности, в качестве устройства связи с операторским пультом.

Разработанный последовательный программируемый интерфейс связи (ППИС) сопрягает канал БВК (соответствует каналу микроЭВМ «Электроника 60») и канал инструментальной миниЭВМ «Электроника 100/25». Линия связи в виде двух витых пар проводов подключается к двум аналогичным платам ППИС, отличающихся канальным стыком. Режим обмена ППИС — двухсторонний асинхронный, символы передаются длиной 5—8 байт. Скорость обмена программируемая, до 9600 бод.

Обеспечена защита от импульсных высокопотенциальных помех и помехоустойчивость по синфазной помехе до 10 В при длине линии связи до 200 м. Наряду с основным элементом БИС КР580ВВ51 в ППИС использованы 10 микросхем серии К589, К555 и К554. Устройство реализовано на печатной плате размером 135×180 мм.

Пульт диалога оператора с вычислительным комплексом системы управления робота

Обучение транспортного робота, т. е. программирование маршрута, сводится к перемещению его по выбранной оператором траектории с описанием действий в характерных точках в интерактивном режиме.

Для снижения психологической нагрузки на оператора и получения информации от БВК в удобной для восприятия форме разработан *многофункциональный выносной пульт* программирования, содержащий символьно-графический дисплей с клавиатурой и двухкоординатную командную рукоятку. Дисплей и клавиатура поддерживают активное общение между оператором и роботом, а рукоятка используется для удобного ручного обучения: направление и скорость движения робота пропорциональны отклонению рукоятки. Портативность выносного пульта позволяет оператору выполнять управление и программирование робота, находясь вне зоны его движения. **Структура пульта** (рис. 2) ориентирована на программную реализацию основных функций по преобразованию информации.

Клавиатура организована по известной схеме матрицы, в узлах которой включены контактные группы клавиш. Опрос матрицы и формирование кодов производится подпрограммой микропроцессора.

Дисплей выполнен на базе бытового телевизора «Электроника 450». Микропроцессор формирует изображения в виде матрицы точек (128 строк при 128 точках в строке) и записывает ее в ОЗУ дисплея, откуда она считывается синхронно с разверткой для модуляции луча. Так как цикл БИС К565РУ3А дотаточно мал для регенерации изображения с частотой 57 Гц при указанном формате,

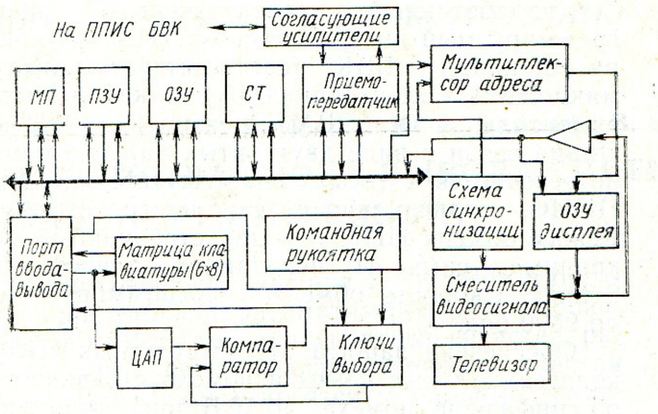


Рис. 2. Структура пульта диалога оператора с БВК

применение параллельного многоразрядного ОЗУ в этом случае нецелесообразно.

Двухкоординатная рукоятка управления роботом построена с использованием потенциометров. Сигнал с них поступает через ключи выбора на вход компаратора. Другой вход компаратора подключен к выходу ЦАП, а выход непосредственно к порту ввода-вывода. Алгоритм аналого-цифрового преобразования с поразрядным уравновешиванием реализуется с помощью программ микропроцессора.

Для периодического опроса всех процессов реального времени в схему включен программируемый таймер.

Пульт выполнен в виде переносного блока (масса — 4,5 кг, габаритные размеры 320×
×210×100 мм при отсоединенной командной рукоятке) и соединяется с роботом кабелем, содержащим линии питания и четыре сигнальных провода. Робот подключается к БВК через последовательный программируемый интерфейс связи (ППИС). Данное устройство как самостоятельная микроЭВМ применяется для управления макетом распределенного объекта управления (стеклоделательной печи) с некоторыми доработками: увеличено число аналоговых входов (до шести) и введены дополнительные порты для управления исполнительными элементами (нагревателями).

Микроконтроллер кассетного накопителя

При обучении транспортного робота, во время сеанса показа маршрута, накапливается информация о перемещениях, остановках и т. д. и директивах, поступающих из подсистемы «диалога с оператором» БВК. Этот массив обрабатывается, и описание маршрута передается в архив рабочих маршрутов, где хранится в долговременной памяти.

Загрузка программного обеспечения системы управления робота требует энергонезависимой памяти. В качестве носителя информации выбрано внешнее запоминающее устройство типа кассетного накопителя на магнитной ленте (КНМЛ). Стандартный контроллер КНМЛ, из состава микроЭВМ «Электроника 60», генерирует высокоинтенсивный поток прерываний. Для снижения нагрузки на центральный процессор БВК (ЦП БВК) по управлению КНМЛ разработан **микропроцессорный контроллер (МК),** передающий функции управления КНМЛ центральному процессору БВК лишь при выходе из строя (рис. 3). Таким образом, к контроллеру были предъявлены повышенные требования по надежности. Функционирование МК контролируется «фоновой» текст-программой, когда МК не занят обслуживанием КНМЛ, и аппаратно, когда он выполняет управление КНМЛ. Были проанализированы и выделены неисправности, приводящие к безвозвратной потере информации. Учитывая, что в большинстве случаев это характерно лишь для КНМЛ в режиме «запись», было решено ограничиться контролем этого режима. Кроме того, возможен контроль от ЦП БВК по времени ответа МК о конце обслуживания КНМЛ.

Другая особенность разработки — организация структуры данных на магнитной ленте. Стандартный формат файлов на ленте для реализации архива неэффективен, так как требует периодической процедуры сжатия. Блочно-индексная структура эффективнее; лента перед использованием размечается для блоков по 128 байт.

Микропроцессорный контроллер разработан на базе микропроцессорного комплекта КР580. Он занимает 512 адресов, отведенных в ЦП БВК под адреса внешних устройств. Основные узлы МК, кроме ПЗУ, программно-

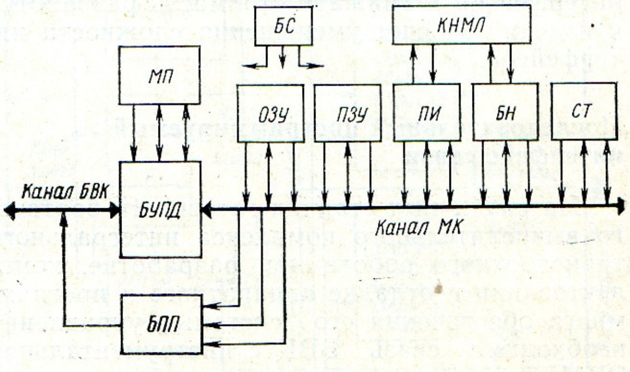


Рис. 3. Структура микропроцессорного контроллера КНМЛ:

БУПД — блок управления прямым доступом; МП — микропроцессор; ОЗУ — оперативное запоминающее устройство емкостью 256 байт; ПЗУ — постоянное запоминающее устройство емкостью 1 Кбат; ПИ — последовательный интерфейс; БН — буфер КНМЛ; СТ — системный таймер, состоящий из двух независимых счетчиков: таймера теста (ТТ) и таймера контроля записи (ТКЗ); БПП — блок приоритетных прерываний; БС — блок синхронизации

доступны как со стороны МП, так и ЦП БВК (см. рис. 3). Обращение к ним ведется, как к ячейкам памяти с соответствующими адресами. Управляющие программы и тест-программа хранятся в ПЗУ. Выполняя программы обслуживания, МП через последовательный интерфейс (ПИ) обменивается данными с КНМЛ. Данные хранятся в ОЗУ МК блоками по 128 байт. В текущий момент времени в ОЗУ может храниться не более одного блока. В ОЗУ также выделены специальные ячейки, которые используются как регистры команды и состояния МК. Обмен информацией с ЦП БВК происходит в режиме «захват шин» (активное устройство — ЦП БВК).

Входные и выходные сигналы управления КНМЛ передаются и принимаются МП через буфер накопителя. Системный таймер (СТ) контролирует время нахождения МК в режимах «тест-программа» и «запись».

БПП выдает прерывания в канал ЦП БВК в случаях отказа, а также в режиме работы под непосредственным управлением ЦП БВК по готовности ПИ к обмену. Кроме того, БПП может выдать прерывание по концу обслуживания запроса ЦП БВК.

БУПД обеспечивает подключение шин МК к каналу ЦП БВК для организации обменов при нормальной работе и отказе МК. В режиме «тест-программа» МП циклически перед началом теста сбрасывает таймер теста (ТТ), опрашивает флаг готовности данных и перехода, в зависимости от флага, либо к тесту, либо к обслуживанию запроса ЦП БВК. Флаг готовности данных показывает, что ЦП БВК заполнил все служебные ячейки ОЗУ. Тип запроса определяется адресом программы обслуживания КНМЛ, который ЦП БВК засылает в одну из служебных ячеек.

Сразу после сброса ТТ начинается отсчет интервала времени выполнения теста. В длительность этого интервала включено также время обмена ЦП БВК с ОЗУ МК при подготовке данных для МК, так как ТТ на время обмена не останавливается.

Если флаг готовности данных установлен, МП считывает начальный адрес программы обслуживания, устанавливает флаг «занято», который ЦП БВК опрашивает каждый раз перед обращением в МК, и переходит к выполнению этой программы. ТТ блокируется сразу, как только на КНМЛ будет подан сигнал включения двигателей лентопротяженного механизма. Запустить ТТ теперь можно, программно сбросив блокировку.

При работе МК в режиме «запись» запускается таймер контроля записи (ТКЗ). Он контролирует время активного состояния следующих сигналов: включение двигателей КНМЛ, включение записи КНМЛ и сигнала готовности передатчика ПСА к приему данных. При отсутствии хотя бы одного из них таймер сбрасывается и блокируется до нового одновременного появления этих сигналов.

Рассмотрим режим «отказ». Под отказом будем понимать одну из следующих ситуаций: МП не выполнил тестовую программу за допустимое время; в режиме «запись» три указанные выше сигнала присутствовали более допустимого времени.

После поступления сигнала «отказ» БПП выдает прерывание в ЦП БВК с соответствующим вектором. Одновременно БУПД переведет МК в режим «принудительный захват», который отличается от режима «захват шин» тем, что все шины МК будут переключены на канал ЦП БВК независимо от состояния сигнала «подтверждение захвата» микропроцессора. Чтобы вернуть систему в исходное состояние (например, после ремонта), нужно подать команду «общий сброс» с ЦП БВК.

Приведенная структура МК хорошо приспособлена для разработки и отладки программного обеспечения. Для этого ПЗУ заменяется на отладочное ОЗУ, к которому разрешен доступ со стороны канала инструментальной микроЭВМ. Последняя связана через ППИС с основной инструментальной мини-ЭВМ «Электроника 100-25». На время отладки МК используется 2К адресов внешних устройств инструментальной микроЭВМ для доступа к отладочному ОЗУ.

Разработанное устройство аппаратно независимо в отношении типа используемого ВЗУ. При замене типа КНМЛ достаточно заменить программное обеспечение МК. Предельная скорость обмена информации с ВЗУ около 20 Кбайт/с.

Предполагается разработка версии МК с микропроцессором, имеющим систему команд ЦП БВК. Это позволит освободить память ЦП БВК от «балласта» резервных программ обслуживания ВЗУ при использовании ПЗУ МК.

ЛИТЕРАТУРА

1. Охоцимский Д. Е., Платонов А. К. и др. Мобильный колесный робот. — В кн.: Управление робототехническими системами и их очувствление. — М.: Наука, 1983.

2. Порозов Б. Ю., Коннов С. Б., Комаров Ю. В., Гудков С. Н. О разработке подвижного робота, управляемого вычислительной машиной. — В кн. Информационные и управляющие системы роботов. — М.: ИПМ — МГУ, 1982.

3. Порозов Б. Ю., Коннов С. Б., Комаров Ю. В., Гудков С. Н. Структура транспортного подвижного робота и анализ отказов в системах. — Тезисы докл. 1 Всесоюзной конф. «Адаптивные роботы—82», г. Нальчик, 1982.

Статья поступила 18 апреля 1984 г.

УДК 007.52:658.286

Ю. В. Чернухин, В. П. Носков, И. А. Каляев, Л. Ж. Усачев,
В. К. Мишкинюк, П. С. Сологуб

МИКРОПРОЦЕССОРНАЯ СИСТЕМА УПРАВЛЕНИЯ ТРАНСПОРТНОГО РОБОТА

Система управления движением транспортного робота на основе микроЭВМ и БИС однородного процессора технологична, удобна в настройке, позволяет осуществлять параллельный ввод информации, обладает повышенной надежности.

Адаптивные промышленные роботы, к поколению которых относятся и транспортные роботы (ТР), должны самостоятельно ориентироваться в нестрого определенной обстановке при условии, когда связь с оператором может быть затруднена. На систему управления ТР возлагаются функции оперативного контроля в реальном масштабе времени без непосредственного участия человека. Такая система может быть построена на основе *однородного процессора,* т. е. процессора, изготовленного в виде *однородной структуры,* позволяющего организовать *параллельный анализ* различных вариантов движения ТР к цели и значительно повысить быстродействие системы управления в целом.

Структурная организация системы управления. Система управления автономного ТР должна содержать (рис. 1) блоки сенсорных датчиков, формирования модели внешней среды, задания цели, навигации и принятия решения. С помощью блока сенсорных датчиков воспринимается информация о рабочей зоне ТР, которая далее используется для построения модели внешней среды и поступает на входы блока принятия решений. Блоки навигации и задания цели служат для определения текущего и целевого положения ТР в рабочей зоне. Ведущую роль в системе играет блок принятия решений, функционирование которого заключается в планировании траектории движения ТР к цели, минуя препятствия, и формировании сигналов отработки найденной траектории.

Использование однородных структур для построения блока принятия решений — новый подход к решению задачи планирования траектории и управления движением автономного транспортного робота в среде с препятствиями. Суть этого подхода заключается в следующем. Рабочая среда ТР разбивается на множество непересекающихся участков, соизмеримых с размерами робота, и каждому такому участку ставится в соответствие вершина графа, ребра которого соединяют вершины, соответствующие соседним участкам. В графе выделяется подмножество вершин — участков положения препятствий в рабочей зоне, а также вершины, соответствующие участкам текущего и целевого положений ТР. Полученная таким образом модель рабочей зоны отображается в состояниях ячеек однородной структуры (ОС), топологически подобной графу. В ОС распространяется волна сигналов из ячейки, соответствующей целевому положению робота. При этом первый сигнал распространения волны, поступивший на ячейку, соответствующую текущему положению ТР, определяет первый шаг кратчайшей траектории движения ТР к цели. После отработки найденного шага исполнительными системами ТР процесс работы ОС повторяется заново с учетом нового положения робота в среде, и так далее до тех пор, пока робот не достигнет цели. Особенность описанной процедуры заключается в том, что в ОС осуществляется параллельный анализ различных вариантов движения ТР к цели. Последнее обстоятельство позволяет добиться существенного выигрыша в быстродействии блока принятия решения, построенного на базе ОС, по сравнению с аналогичными блоками, использующими последовательные ЭВМ.

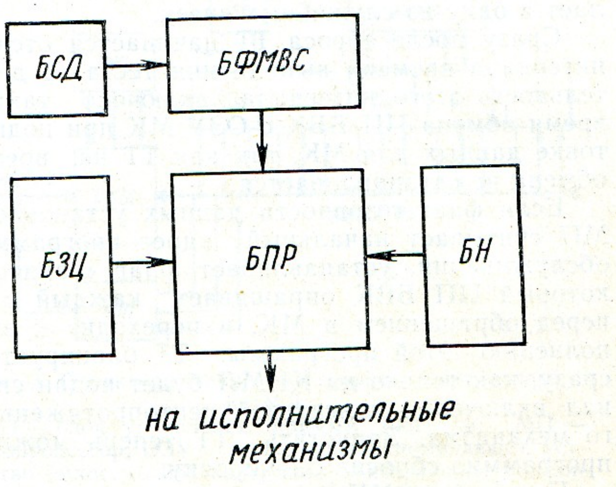


Рис. 1. Структурная схема системы управления: БСД — блок сенсорных датчиков; БФМВС — блок формирования модели внешней среды; БЗЦ — блок задания цели; БПР — блок принятия решения; БН — блок навигации

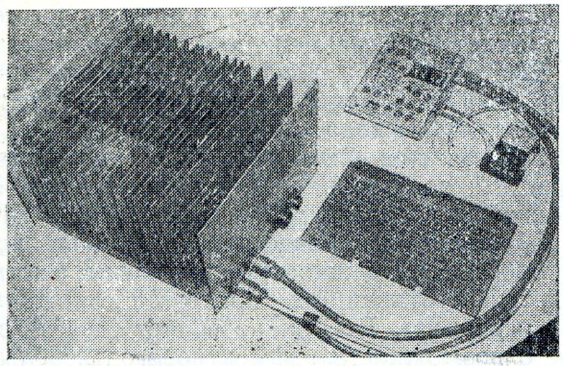


Рис. 2. Макет блока принятия решения в виде однородной структуры

Результаты экспериментальных исследований. Для практической проверки предложенных теоретических принципов разработан и изготовлен макет системы управления автономного ТР. Основу системы составляет специализированный процессор, выполняющий функции блока принятия решений и изготовленный в виде однородной структуры (рис. 2). Процессор содержит 448 ячеек, построенных на базе микросхем серии К164. Ультразвуковая система навигации обеспечивает определение текущих координат ТР в рабочей зоне размерами 50×50 м^2 (информация об ориентации ТР поступает в систему управления от гироскопического компаса). В качестве блока задания цели в системе используется пульт.

Информация о рабочей зоне робота может поступать в систему тремя способами:

набирается с пульта перед началом функционирования робота;

робот самостоятельно накапливает информацию о среде в запоминающем устройстве с помощью тактильных датчиков, установленных на корпусе;

информация поступает от лазерного дальномера.

В последнем случае в систему необходимо ввести специализированный предпроцессор, выполняющий функции блока формирования модели внешней среды.

Функционирует макет системы управления следующим образом. План рабочей зоны, накопленный в процессе функционирования ТР или занесенный заранее, хранится в оперативной памяти системы управления в виде матрицы двоичных признаков, определяющих проходимость соответствующих дискретных участков рабочей зоны. Цель движения ТР задается с пульта в виде координат конечного участка. В каждом цикле работы системы управления информация о препятствиях и цели отображается в состояниях ячеек процессора с учетом текущего положения ТР. В результате в процессоре выделяется текущий шаг кратчайшей траектории движения ТР к цели, который отрабатывается исполнительными механизмами, и в итоге ТР выходит к цели по кратчайшей траектории, минуя препятствия. Время выработки команд управления на исполнительные устройства ТР не превышает 0,5 с. Такое быстродействие позволяет использовать систему с однородным процессором для управления роботом в реальном масштабе времени. Система имеет массу около 12 кг, габаритные размеры — $360\times320\times240$ мм, позволяющие размещать ее на транспортной тележке.

Вопросы микропроцессорной реализации системы управления. Анализ функций, выполняемых блоками формирования модели внешней среды, задания цели и навигации, позволили сделать вывод о возможности их реализации с помощью микроЭВМ типа «Электроника 60». В то же время высокая однородность и простота схемы процессора позволяют изготовить его в виде БИС. Таким образом, *систему управления автономного ТР можно выполнить в виде комплекса, состоящего из стандартной микроЭВМ и БИС однородного процессора.*

Структура однородного процессора, топологически подобна графу модели внешней среды робота. От архитектуры графа будет в значительной степени зависеть степень приближения получаемой траектории к кратчайшей. Проведенные исследования показали, что наиболее предпочтительной является концент-

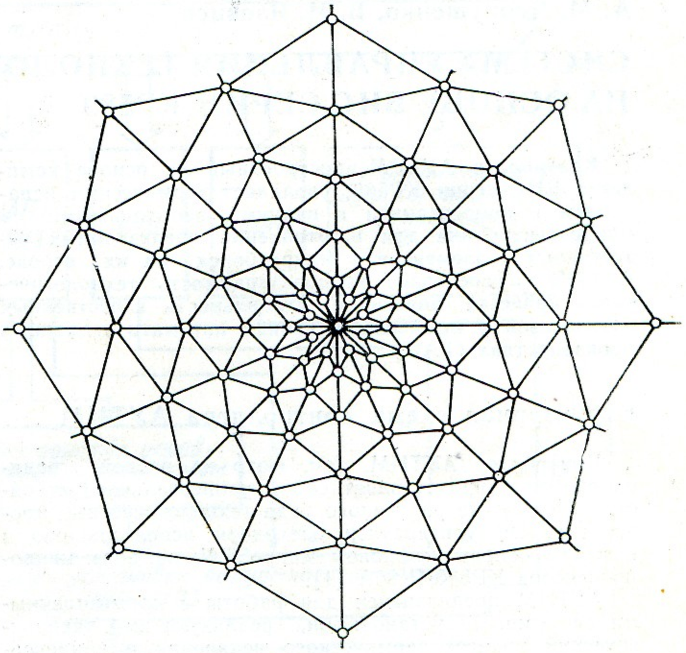


Рис. 3. Архитектура управляющей структуры

рическая архитектура с гексагональными связями между вершинами (рис. 3), причем текущее положение робота жестко связано с центральной вершиной графа. Такой граф позволяет моделировать рабочую зону с переменной дискретностью. При этом близлежащие области будут представляться в виде совокупности большего числа вершин графа, т. е. будут моделироваться более точно, а более удаленные области — в виде меньшего числа вершин графа, т. е. будут моделироваться более грубо. Необходимо отметить, что такая модель адекватна форме восприятия рабочей среды бортовыми сенсорными системами ТР, например, лазерными дальномерами, у которых разрешающая способность падает с увеличением дальности обзора. Кроме того, представление рабочей зоны в виде концентрического графа требует ее разбиения на меньшее число дискретных участков, чем представление в виде регулярного графа. Последнее обстоятельство позволяет уменьшить аппаратные затраты при реализации графа в ячейках процессора.

Проведенные исследования показали, что для реализации однородного процессора, содержащего 448 ячеек и реализующего концентрический граф, требуется около 20 000 транзисторов. Такое число элементов уже сейчас может быть размещено на одном кристалле БИС. При этом для подключения внешних устройств БИС однородного процессора должна иметь 64 вывода.

Заключение. Система управления движением ТР, построенная на базе комплекса, состоящего из стандартной микроЭВМ и БИС однородного спецпроцессора, обладает компактностью, надежностью, высоким быстродействием, отличается простотой изготовления и ремонта. Все это позволяет сделать вывод о возможности эффективного использования такой системы для обслуживания роботов, автономно функционирующих в естественной среде в реальном масштабе времени.

ЛИТЕРАТУРА

1. Каляев А. В., Носков В. П., Чернухин Ю. В. Алгоритм управляющей структуры транспортного робота. — Известия АН СССР. Сер. Техническая кибернетика, 1980, № 4, с. 64—72.

2. Каляев А. В., Каляев И. А., Носков В. П., Чернухин Ю. В. Однородные управляющие структуры адаптивных роботов. — Электронная промышленность, 1981, № 10, с. 19—23.

Статья поступила 23 апреля 1984 г.

МИКРОПРОЦЕССОРЫ В ПРОИЗВОДСТВЕННОМ ОБОРУДОВАНИИ

УДК 621.3.049.77

Н. П. Давыдов, Ж. К. Давыдова, Г. Т. Мишин, А. П. Савченко, А. М. Чернушенко, В. И. Язовцев

СИСТЕМА УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМИ ПРОЦЕССАМИ НА ОСНОВЕ БИС СЕРИИ КР580

Контроллер АУТИМ, построенный на основе комплекта БИС серии КР580, управляет термическим испарением и конденсацией в вакууме, необходимыми в микроэлектронике для изготовления прецизионных резистивных элементов и микросборок на их основе. АУТИМ обеспечивает воспроизводимость технологического процесса, повышает стабильность качественных показателей и применим в гибких автоматизированных производствах (ГАП).

Структурная схема контроллера АУТИМ

Контроллер АУТИМ (см. фото на первой странице вкладки) представляет собой универсальное устройство управления различного рода технологическими процессами. Он универсален благодаря использованию в качестве основы цифровой микроЭВМ на базе микропроцессора КР580ИК80А [1].

АУТИМ предназначен для работы с промышленными вакуумными установками, реализующими технологический процесс термического испарения и конденсации в вакууме, используемый в микроэлектронике для изготовления тонкопленочных резистивных элементов. Структурная схема АУТИМ (рис. 1) предопределена характеристиками производственного объекта (вакуумной установки промышленного образца типа УВН-71П3) и типом микропроцессора.

Контроллер АУТИМ выполнен на базе микропроцессора КР580ИК80А [2], но это не означает, что другие микропроцессоры непригодны. Выбор микропроцессора серии КР580 произведен на основании анализа, выполненного на начальном этапе разработки.

Основные доводы — у микропроцессора КР580ИК80А законченное архитектурное оформление (это позволяет не тратить средств на разработку собственного микропроцессора); система команд микропроцессора КР580ИК80А сформирована в виде языка, условно называемого ассемблером (это позволяет не тратить средств на разработку языка); тактовая частота синхронизации микропроцессора соизмерима с тактовой частотой процессоров других серий.

В ходе предварительной проработки рассмотрена возможность использования в качестве основы серийной универсальной микроЭВМ, например, «Электроника 60». Сравнительный анализ структурных схем контроллера и микроЭВМ привел к выводу о целесообразности разработки самостоятельной конструкции:

в структурной схеме контроллера большое количество модулей определяют персональную ориентацию контроллера: модули программно-управляемого цифрового измерительного устройства (ПУЦИС), сопряжения контроллера с производственным объектом, интерфейсные и т. д.; условия работы цифровых вычислительных устройств в системах автоматического управления технологическими процессами, которые воспринимаются нами как непрерывные, значительно отличаются от условий работы универсальных вычислительных устройств. Это налагает отпечаток на конструктивные решения. Так, эксплуатация варианта контроллера, показанного на фотоснимке (на первой странице вкладки), привела к коренной переработке конструкции передней панели. Это несложно при модульном варианте архитектуры и, наоборот, сопряжено со значительными трудностями при использовании универсальной микроЭВМ.

Системная магистраль, состоящая из шин адресов, данных и управляющих сигналов, должна обладать высокой нагрузочной способностью для подключения сравнительно большого числа рабочих модулей. Нагрузочная способность контроллера повышается при помощи микросхем К589АП16 и К589АП26.

Работу системной магистрали организует модуль центрального процессора (ЦП) — на базе БИС КР580ИК80А. Схемное решение модуля ЦП в основном типовое [1]. Некоторые изменения внесены в блок управления и синхронизации (для более устойчивой работы всего контроллера).

Системная магистраль позволяет к универсальной микроЭВМ подключить устройства сопряжения (УСО), необходимые для организации автоматически управляющей системы. Набор устройств сопряжения и соответствующая технологическому процессу программа управления, помещаемая в сменных микросхемах блока ППЗУ, придают персональную ориентацию (уникальность) устройству управления.

Модуль устройства управления пультом (УУП) реализован на 27 микросхемах серий К155 и К589. УУП управляет запуском и остановом программы, прямым доступом к памяти, тестированием устройств сопряжения и визуальным контролем выполнения программы.

В отдельный модуль выделены схема прерываний и дешифратор (СПиД) на микросхемах серии К589 и К155ИД3 или КР580ВН59. Аппаратное решение схемы дешифратора позволяет применять блоки памяти емкостью до 1 Кбайт. Максимальная емкость памяти (ОЗУ или ПЗУ) при использовании данного дешифратора — 16 Кбайт. Благодаря дешифратору количество блоков памяти в 1 Кбайт легко увеличить или уменьшить без дополнительной их привязки к каким-либо определенным адресам, а адресацию памяти удобно изменять для решения различных задач.

Модуль оперативного запоминающего устройства (ОЗУ) на основе микросхем К565РУ2А статического типа имеет емкость 4 Кбайт. Хранение промежуточных результатов при арифметическо-логической обработке данных (если система работает в режиме реального времени) требует не более 1 Кбайт памяти. Остальная часть памяти предназначена для сервисных программ, в частности, для хранения данных о процессе и построения различных графиков после его окончания

Модуль перепрограммируемого запоминающего устройства (ППЗУ) выполнен на основе микросхем типа К573РФ14. Это дает возможность перезаписи информации после стирания ультрафиолетовыми лучами. Основное назначение модуля — хранение рабочей программы и библиотеки программ математических операций. Рабочая программа и библиотека математических операций записаны в парах микросхем побайтно: старший байт — в одной, младший — в другой. Архитектура модуля ППЗУ — типовая [1]. Объем памяти можно изменять. Максимально возможный объем памяти — 3 Кбайт, используемый — 2 Кбайт.

Модуль управления вибробункером (УУВ) на базе микросхем К155ИЕ8 служит для программно-управляемого деления частоты 200 кГц, формируемой из тактовой частоты микропроцессорного модуля в модуле СПиД, и выдачи управляющего воздействия $U_б$ на вибробункер (дозирующее устройство) в подколпачном устройстве вакуумной установки.

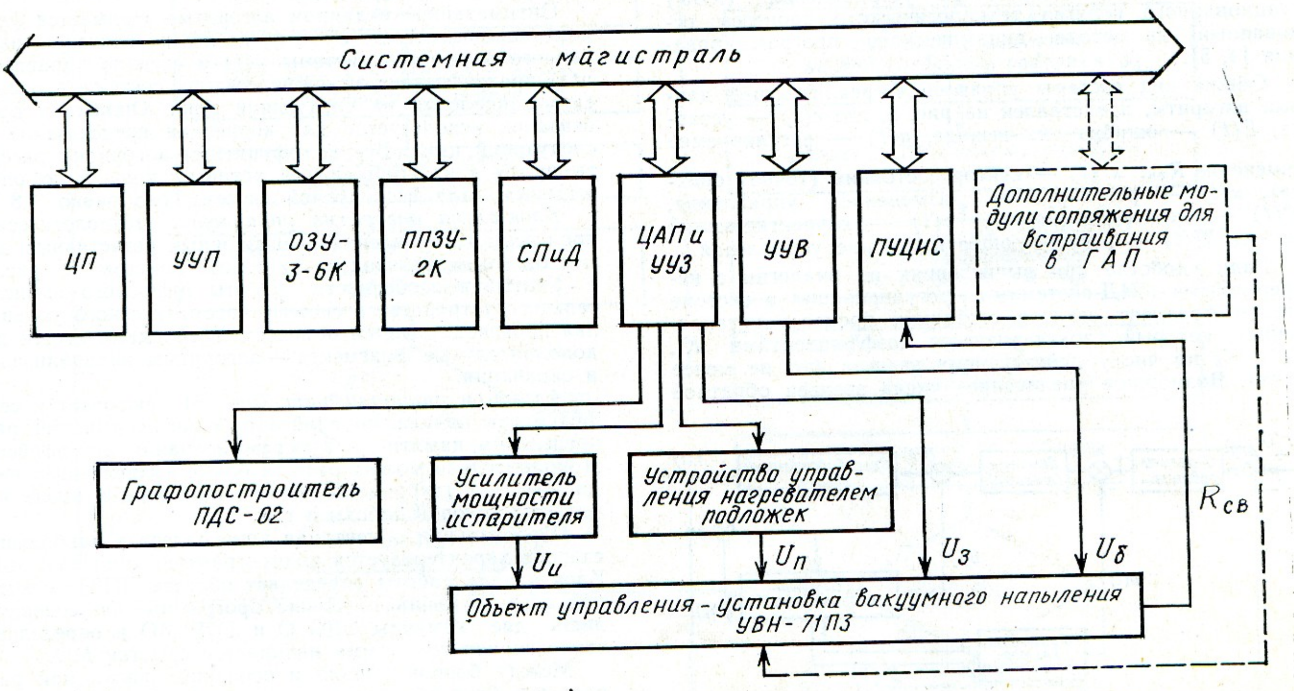


Рис. 1. Структурная схема контроллера АУТИМ

Модуль цифро-аналогового преобразователя (ЦАП) и устройства управления заслонкой (УУЗ) состоит из четырех ЦАП, на основе микросхем К572ПА1В и К140УД6. Эти преобразователи выводят аналоговое управляющее воздействие на испаритель $U_и$, нагреватель подложки $U_п$, а также на двухкоординатный графопостроитель для воспроизведения графиков, характеризующих технологический процесс. Цепь управления заслонкой $U_з$ гальванически развязана со схемой контроллера.

Схема контроллера, реализованная на практике [3], конечно, не свободна от недостатков. Пути развития архитектуры контроллеров для сложных технологических процессов наметились в результате экспериментальной эксплуатации АУТИМа:

Целесообразно, видимо, говорить о *цеховом контроллере* для управления производственным объектом по заранее разработанным и испытанным модели и программе. Он должен быть возможно более прост, иметь минимум ОЗУ и ППЗУ, удобный пульт с минимальным количеством органов управления и, с этой точки зрения, более всего походить на одноплатную микроЭВМ.

В то же время совершенно необходим *лабораторный контроллер,* с модифицированной архитектурой, ОЗУ и ППЗУ большого объема, встроенным графическим дисплеем и возможностью сопряжения со стандартным интерфейсом ЭВМ. Его назначение — исследование технологических процессов с целью их автоматизации, определение различного рода постоянных коэффициентов, технологических режимов, обработка фрагментов программ и алгоритмов и т. п.

Алгоритм управления технологическим процессом

Одним из наиболее удачных с практической точки зрения алгоритмов, применяемых для управления исследуемым технологическим процессом, следует считать алгоритм с конечным временем установления (КВУ) стационарного режима технологического процесса, основанный на методе динамического программирования [4, 5].

Общий вид системы управления, реализующей данный алгоритм, представлен на рис. 2, где $r(t)$ — уставка, $E(t)$ — ошибка на выходе, $y(t)$ — регулируемое изменение $R_{св}$, $\overset{\vee}{x}(t)$ — вектор состояния (точная оценка), $u(t)$ — регулируемое изменение управления, $y°(t)$ — априорная оценка, $b(t)$ — корректирующий сигнал, $W(t)$ — регулирующее изменение управления.

Ради удобства при вычислениях по аналогии с непрерывными ПИД-системами регулирования в системе введен фрагмент «вектор состояния процесса установления», который характеризуется коэффициентами K_1, K_2 и K_3 по числу действующих звеньев (но не менее двух). Возможное увеличение числа звеньев обратной

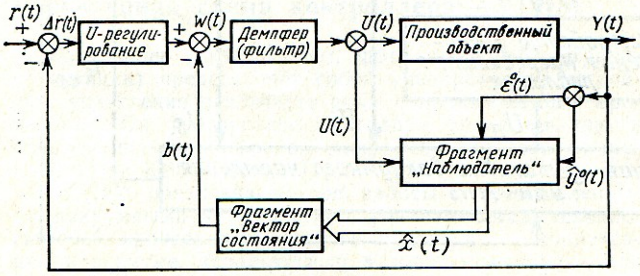


Рис. 2. Общий вид системы управления

Таблица 1

Пример загрузки исходных данных в шестнадцатеричной системе на языке ассемблер

Метка	Адрес	Наименование параметра	Вводимая величина в шестнадцатеричной системе
ADR 50	0C32	Начальный режим испарителя	80
51	0C34	Коэффициент настройки	40
52	0C36	Коэффициент модели	60
53	0C38	Коэффициент модели	20
54	0C3A	Коэффициент модели	6A
55	0C3C	Коэффициент алгоритма	08
56	0C3E	Коэффициент алгоритма	04
57	0C40	Коэффициент чувствительности подложки	1C
58	0C42	Коэффициент чувствительности испарителя	18
59	0C44	Коэффициент чувствительности вибробункера	10
60	0C46	Постоянная удельного сопротивления материала	42
61	0C48	Постоянная коэффициента формы свидетеля	36
62	0C4A	Коэффициент демпфера	14
63	0C4C	Конечное сопротивление свидетеля	10
64	0C4E	Максимальное число рабочих циклов	FF
65	0C50	Нулевая уставка	258

связи позволяет (при корректном определении коэффициентов K_1, K_2 и K_3), улучшить или даже оптимизировать реакцию контроллера.

Оптимизацией в данном алгоритме занимается фрагмент «наблюдатель». Этот фрагмент по значению измеренного параметра системы $y(t)$ и сигнала управления $u(t)$ предсказывает значение вектора состояния процесса установления на следующий цикл. Ошибка в предсказании используется для коррекции предсказания на следующий цикл. Все коэффициенты алгоритма рассчитываются так, чтобы время установления было оптимальным (для исследуемой системы оно равно ~ 8 с).

Блок-схема алгоритма управления технологическим процессом изготовления прецизионных резистивных элементов и микросборок на их основе показана на рис. 3.

Учитывая особенности работы цифрового вычислительного устройства в системе автоматического управления производственным объектом, необходимо ввести два дополнительные фрагмента — алгоритмы инициализации и окончания.

Алгоритм инициализации (рис. 4) включает в себя фрагменты начала, порядка обработки прерываний, распределения памяти и программирования интерфейсов. Дополнительно можно отнести блоки вычисления и ввода констант техпроцесса, блок вычисления и ввода начальных условий работы и т. д.

Персональная ориентация контроллера в наибольшей степени характеризуется подпрограммой инициализации. Блок начала работы определяет область ППЗУ, с которой следует начинать чтение программы. Он содержит лишь две команды ORG O и JMP MO и определяет, что основная программа начинается с метки МО.

Между блоком начала и основной программой расположен блок обработки прерываний. По прерыванию данная программа командой JMP M10 возвращает процессор на начало рабочей программы.

Блок загрузки исходных данных регламентирует порядок и количество данных, вводимых с переднего пульта перед началом процесса (табл. 1). Программа, соответствующая блоку загрузки, с необходимыми комментариями, приведена в табл. 2. Если число вводимых данных меньше 25, то по остальным адресам набираются нули. Сигнал записи двадцать пятого исходного данного одновременно является сигналом продолжения выполнения программы. Заметим, что и предельное число исходных данных зависит от типа процесса и от степени проработки его технологии. В реализованном варианте вводятся шестнадцать данных (см. табл. 1), из которых два характеризуют технологический режим, шесть — граничные условия протекания процесса, а восемь — коэффициенты модели, алгоритма и производственного объекта. Многочисленность вводимых данных объясняется оригинальностью алгоритма и отсутствием регулярных методик определения коэффициентов. После подробной проработки процесса должна быть написана сервисная программа определения постоянных по результатам выполнения серии предварительных производственных процессов, а количество вводимых данных должно уменьшиться до 3—5.

Существующие измерительные приборы для достижения точности выполняются многодиапазонными с механическими переключателями диапазонов, а потому несовместимыми с микропроцессорными устройствами (МПУ).

Программно-управляемый цифровой измеритель сопротивлений (ПУЦИС) и специальная программа измерителя «*М* изм.» были разработаны для обеспечения возможности считывания результата измерения.

ПУЦИС (рис. 5) — специальный, многопредельный, амплитудно-цифровой преобразователь, выполненный по схеме двойного интегрирования [6].

Измеряемое сопротивление $R_{изм}$ включено в интегрирующую цепь, характеристика которой определяет время, необходимое выходному напряжению интегратора для достижения порога срабатывания компаратора. Эта первая интегрирующая цепь определяет время счета программно-управляемого счетчика импульсов, результат которого фактически и является результатом измерения сопротивления. Управляет измерителем интерфейсное устройство, связанное соответствующими шинами с системной магистралью микроЭВМ.

При сравнительной простоте (27 микросхем и 5 транзисторов) ПУЦИС измеряет сопротивление в пределах 1,0 КОм — 3,8 МОм с точностью 1%. Амплитудно-цифровой преобразователь устройства имеет двенадцать значащих разрядов и разряд переполнения, кото-

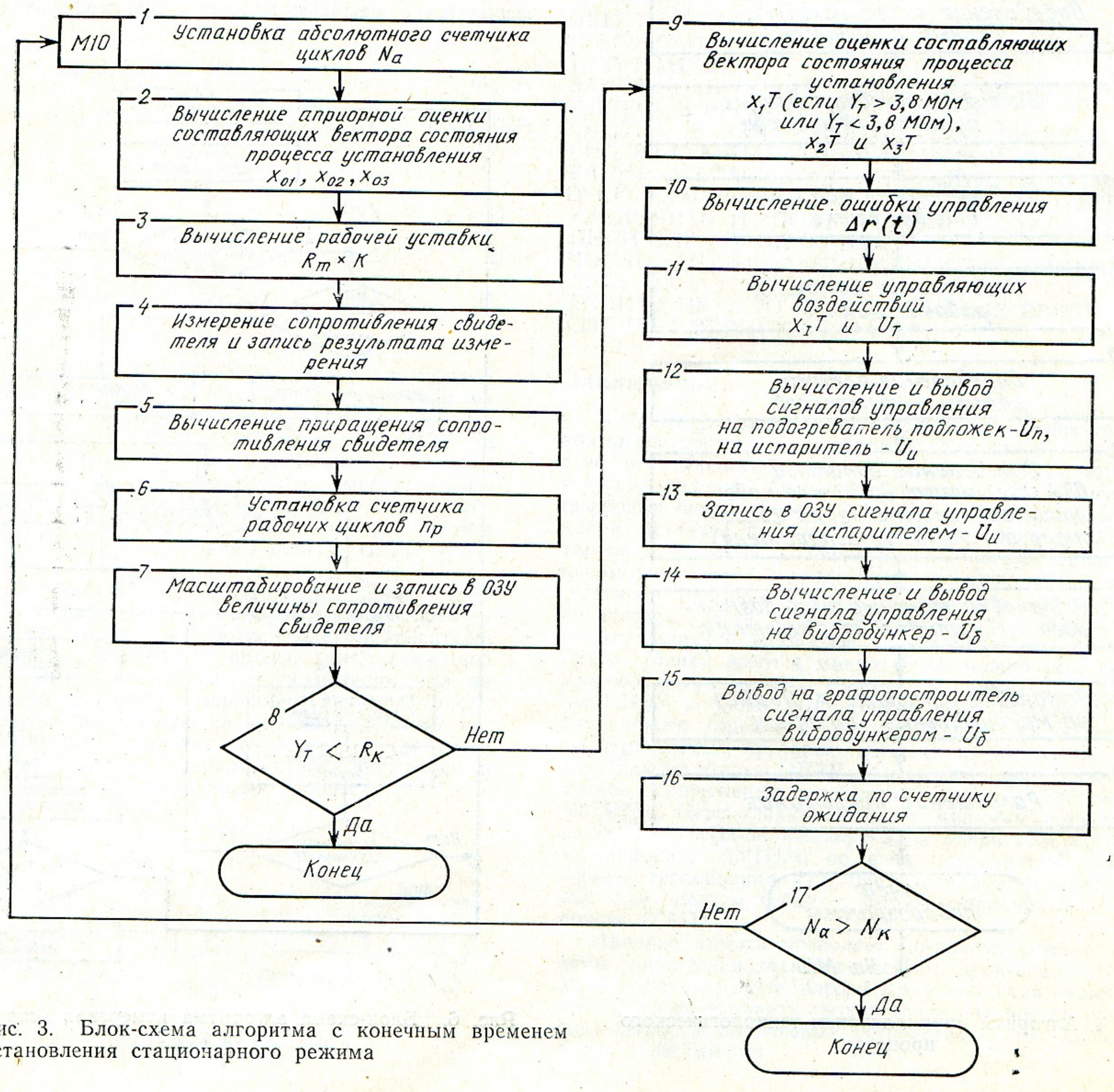


Рис. 3. Блок-схема алгоритма с конечным временем установления стационарного режима

рый может считаться значащим. Запуск измерителя, считывание, интерпретация результатов измерения и останов выполняет специальный алгоритм (рис. 6). Его основная задача — выбор номера переключателя диапа-

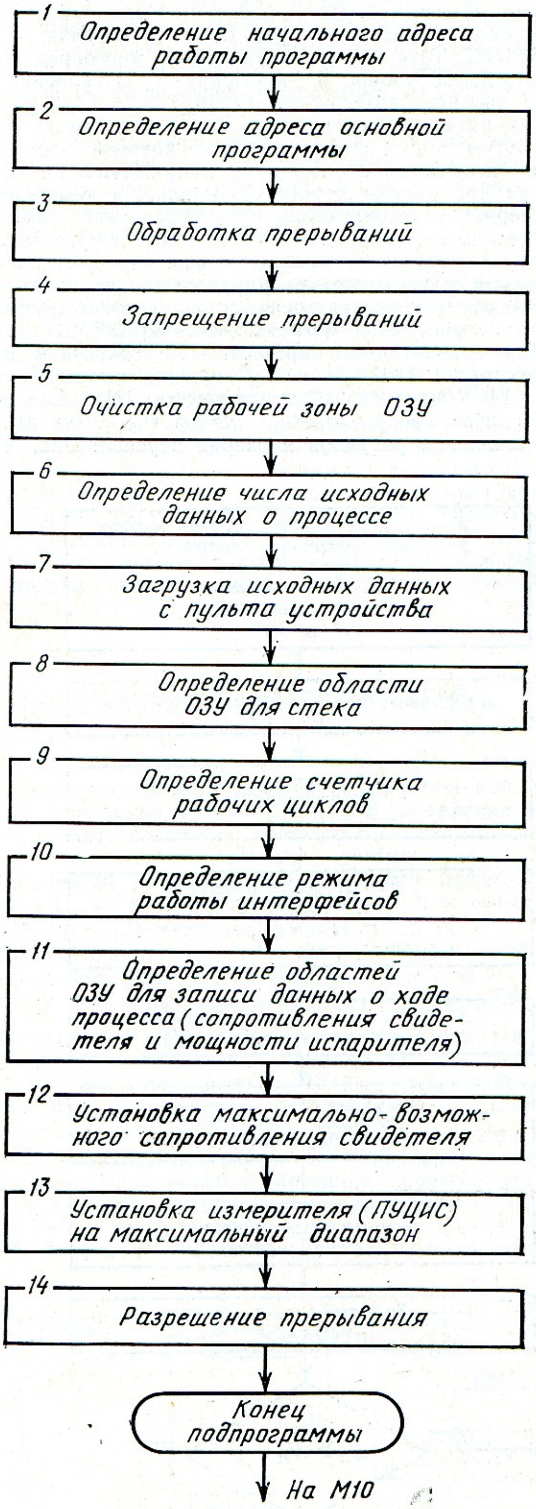


Рис. 4. Алгоритм инициализации технологического процесса

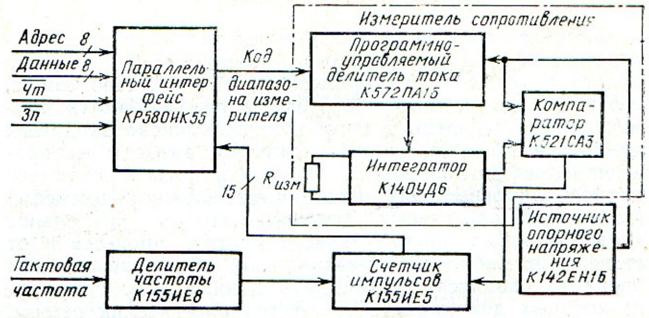


Рис. 5. Структурная схема программно-управляемого цифрового измерителя сопротивления (ПУЦИС)

зона, при котором результат измерений размещается в предпочтительной зоне шкалы АЦП (рис. 7). Дополнительно она преобразует номер диапазона и полученное в АЦП цифровое значение в величину сопротивления в масштабе приблизительно 1:128 и записывает результат измерения в специально отведенную ячейку ОЗУ.

Величины выходных воздействий формируются и выводятся с помощью подпрограммы модификации уп-

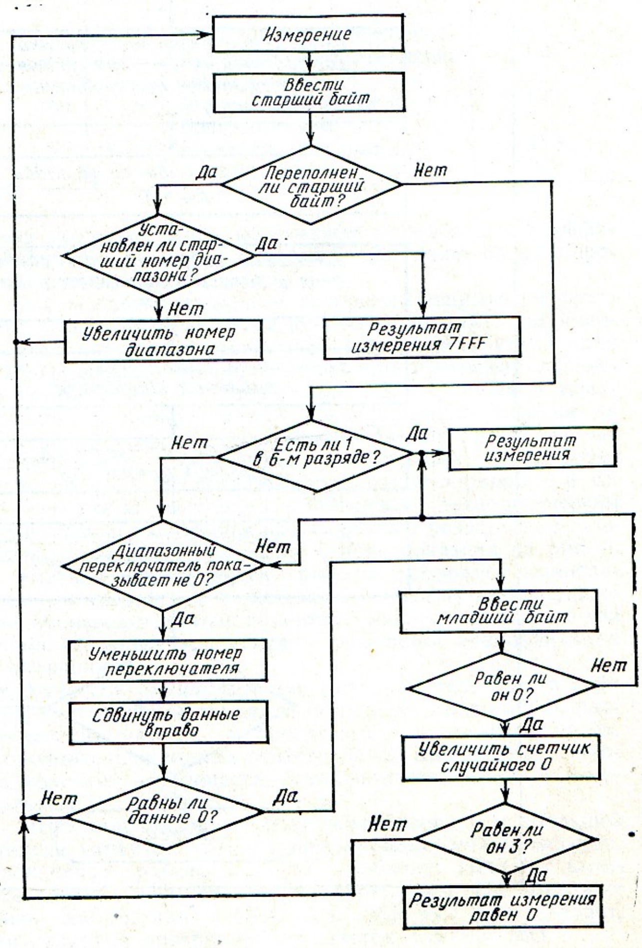


Рис. 6. Блок-схема алгоритма измерения сопротивления $R_{св}$

Таблица 2

Программа загрузки

13:	0018	21320C	LXI	H, ADR50	ЗАГРУЗИТЬ В ПАРУ РЕГИСТРОВ H ADR50 — АДРЕС НАЧАЛА ИСХОДНЫХ ДАННЫХ
14:	001B	1619	MVI	D, 19H	ОПРЕДЕЛИТЬ, ЧТО МАКСИМАЛЬНОЕ ЧИСЛО ИСХОДНЫХ ДАННЫХ=25 (19H)
15:	001D	1E03	M2: MVI	E, 03H	
16:	001F	DBFC	IN	OFCH	ВВЕСТИ В АККУМУЛЯТОР СОДЕРЖИМОЕ РЕГИСТРА, КОТОРОМУ АППАРАТНЫМ СПОСОБОМ ПРИСВОЕН НОМЕР OFCH. ДВА МЛАДШИХ БИТА ЭТОГО РЕГИСТРА УСТАНАВЛИВАЮТСЯ В 1 ПРИ НАЖАТИИ КНОПКИ «ВВОД»
14:	0021	AB	XRA	E	ЛОГИЧЕСКОЕ ИСКЛЮЧАЮЩЕЕ «ИЛИ» МЕЖДУ РЕГИСТРОМ E И АККУМУЛЯТОРОМ
18:	0022	C21D00	JNZ	M2	ЕСЛИ A НЕ РАВНО E, ТО ЕСТЬ КНОПКА ВВОДА НЕ НАЖАТА, ВОЗВРАЩАЕМСЯ НА M2
19:	0025	DBFD	IN	OFDH	ПОСЛЕ НАЖАТИЯ КНОПКИ ВВЕСТИ В АККУМУЛЯТОР СОДЕРЖИМОЕ МЛАДШЕГО БАЙТА ТУМБЛЕРНОГО РЕГИСТРА ПЕРЕДНЕЙ ПАНЕЛИ
20:	0027	77	MOV	M, A	ЗАПОМНИТЬ СОДЕРЖИМОЕ АККУМУЛЯТОРА, ЗАПИСАВ ЕГО В ЯЧЕЙКУ ПО АДРЕСУ, ЗАПИСАННОМУ В ПАРЕ РЕГИСТРОВ H
21:	0028	23	INX	H	УВЕЛИЧИТЬ СОДЕРЖИМОЕ ПАРЫ РЕГИСТРОВ H НА 1
22:	0029	DBFE	IN	OFEH	ВВЕСТИ В АККУМУЛЯТОР СОДЕРЖИМОЕ СТАРШЕГО БАЙТА ТУМБЛЕРНОГО РЕГИСТРА ПЕРЕДНЕЙ ПАНЕЛИ
23:	002B	77	MOV	M, A	ЗАПОМНИТЬ СОДЕРЖИМОЕ АККУМУЛЯТОРА ПО АДРЕСУ В РЕГИСТРОВОЙ ПАРЕ H
24:	002C	DBFF	IN	OFFH	ВВЕСТИ В АККУМУЛЯТОР СОДЕРЖИМОЕ РЕГИСТРА С АППАРАТНЫМ НОМЕРОМ FF. ЭТА ОПЕРАЦИЯ СБРАСЫВАЕТ ЕДИНИЦЫ В РЕГИСТРЕ OFCH, РАЗРЕШАЯ ВВОД СЛЕДУЮЩЕГО ДАННОГО
25:	002E	23	INX	H	УВЕЛИЧИТЬ H, ПОДГОТОВИТЬ АДРЕС ДЛЯ ЗАПИСИ СЛЕДУЮЩЕГО БАЙТА
26:	002F	15	DCR	D	ЧИСЛО НЕ ВВЕДЕННЫХ ДАННЫХ УМЕНЬШИЛОСЬ НА 1
27:	0030	C21D00	JNZ	M2	ЕСЛИ D НЕ 0, ТО НЕ ВСЕ ДАННЫЕ ВВЕДЕНЫ И СЛЕДУЕТ ВЕРНУТЬСЯ НА M2

равляющего сигнала (ЕМУС). При этом подпрограмма ЕМУС определяет величину выходного воздействия на конкретный регулирующий механизм, выводит его на соответствующий интерфейс и записывает в ОЗУ по программному адресу (100). Реализованная система рассчитана на однобайтный выход, т. е. имеет 256 фиксированных уровней выходного сигнала от 0 до 255. Отрицательные сигналы управления интерпретируются программой как нулевые, сигналы больше OFFH (255) — как 255.

По алгоритму окончания (рис. 8) устройства системы последовательно выключаются, а информация обрабатывается и фиксируется на соответствующие носители.

Программа останавливает процесс по достижению свидетелем наперед заданного сопротивления или по числу заданных циклов управления. После останова программа выводит данные на двухкоординатный самописец, который фиксирует графики (в том числе многоцветные) изменения сопротивления свидетеля и технологических режимов во время процесса.



Рис. 7. Шкала АЦП ПУЦИСа

Заключение

Аппаратная часть и программное обеспечение в комплексе составляют цифровой контроллер — устройство, позволяющее по-новому решать сложные технологические задачи. Основное принципиальное отличие цифровых контроллеров от их аналоговых предшественников (наличие памяти) — революционирующий фактор по отношению и к конструкции контроллеров, и к идеям управления технологическими процессами.

Именно наличие памяти позволило унифицировать аппаратную часть, сосредоточив все сведения о конкретном процессе и способе управления им в блоках ППЗУ. Меняя всего 4 микросхемы, можно реализовать разнообразные методы управления. Заменив микросхемы ППЗУ и блоки УСО, можно использовать АУТИМ для гибкого управления производственными объектами, где измеряемым критерием служит сопротивление изделия. Замена блока ППЗУ измерителя сопротивления и блоков сопряжения расширяет сферу применения АУТИМа. Переналадка занимает минимум времени и не требует высокой квалификации оператора. Устройства, подобные АУТИМ, по всем показателям удовлетворяют требованиям [7], предъявляемым к контроллерам для систем гибких автоматизированных производств (ГАП).

Наличие памяти позволяет по-новому решать проблемы управления технологическими процессами. В реализованном варианте алгоритма управления, например, можно прогнозировать результат измерения на следующем цикле. В аналоговых устройствах это принципиально недостижимо.

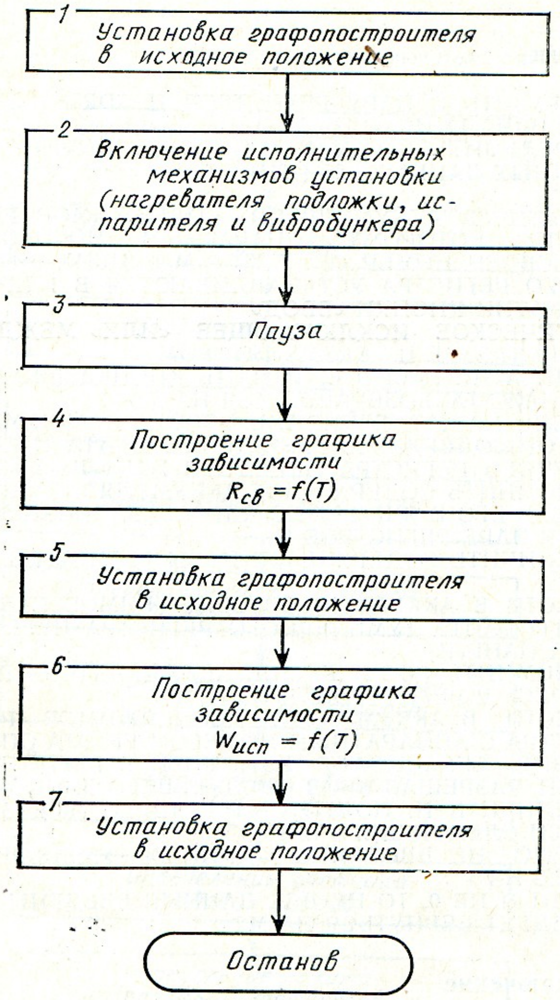


Рис. 8. Алгоритм окончания технологического процесса

С появлением цифровых контроллеров коренным образом меняется идеология технологов. Принцип «поддерживать технологические режимы на постоянном уровне» сменится принципом «гибко менять технологические режимы, чтобы получить заданное качество изделия».

Практика эксплуатации реализованного варианта системы АУТИМ выявила и нерешенные проблемы: отсутствие измерителей, датчиков и исполнительных механизмов, совместимых с микропроцессорными устройствами управления. Предстоит, видимо, исследовать и решить вопросы о принципиальной совместимости высоковольтных силовых устройств на производственных объектах и микропроцессорных устройств управления и о защите от импульсных помех силовых установок. Эти импульсные помехи много больше уровня срабатывания типовых пороговых устройств на микросхемах. Значительные трудности создают вторичные низковольтные, мощные источники питания, которые по габаритам и весу ощутимо превосходят все остальные части конструкции, оставаясь при этом наименее надежной ее частью.

Цифровые контроллеры для сложных технологических процессов в ГАП по своим возможностям не имеют конкурентов среди других технических средств, а по стоимости соизмеримы с ними, поэтому вопрос о их преимущественном развитии уже решен практикой.

ЛИТЕРАТУРА

1. Зеленко Г. В., Панов В. В., Попов С. Н. Радиолюбителю о микропроцессорах и микроЭВМ. — Радио, 1982, № 9, с. 32—36; 1983, № 2, с. 40—43; № 3, с. 31—34.

2. Микропроцессорные комплекты интегральных схем/Под редакцией А. А. Васенкова, В. А. Шахнова. — М.: Радио и связь, 1982, с. 7—13.

3. Давыдов Н. П., Мишин Г. Т., Язовцев В. И. Управление качеством процесса термического испарения и конденсации с помощью микропроцессорных устройств. — В кн.: Микроминиатюризация радиоэлектронных устройств. РРТИ, 1983, с. 9—14. (Межвуз. сб. науч. тр.).

4. Ауслендер Д. М., Такахаси Я., Томидзука М. Применение микропроцессоров для прямого цифрового управления процессами и алгоритмы для контроллеров на микропроцессорах. — ТИИЭР, 1978, т. 66, № 2, с. 113—124.

5. Tomizuka M., Auslender D. M., Takahashi Y. Simple Finite-Time Settling Control and Manipulated—Variable Softening for reverse Reaction Overshoot, and Oscillatory Processes. — Transactions of the ASME, Jornal of Dynamic System, Measurement and Control, 1978, March, vol. 100, p. 50—58.

6. Гнетек Ю. Р. Справочник по цифро-аналоговым и аналого-цифровым преобразователям. — М.: Радио и связь, 1982. — 194 с.

7. Романов А. К. Микропроцессорная техника и автоматизация народного хозяйства. — Микропроцессорные средства и системы, 1984, № 1, с. 4.

Статья поступила 18 апреля 1984 г.

РЖ ВИНИТИ

Удобный интерфейс для пользователей-непрофессионалов. A friendly interface for the lay user. Watts R. A. «Int. Conf. Man/Mach. Syst., Manchester, 6-9 Jule, 1982». London; New York (англ.)

Перечисляются требования, предъявляемые к организации диалога с ЭВМ, рассчитанного на непрофессионального пользователя. Рассматриваются преимущества и недостатки построения интерфейса на базе подмножества естественного языка, инициируемого ЭВМ диалога, совокупности «меню», запросов с подсказками и т. д. Приводятся некоторые рекомендации для разработчиков систем, рассчитанных на случайных пользователей. [Nat. Computing Centre, Великобритания].

Направления развития, технология и стратегия конструирования микроЭВМ. Microcomputers: trends, technologies, and design strategies. Murphy Bernard T. «IEEE J. Solid-State Circuits», 1983, 18, № 3 (англ.)

Анализируются перспективы развития отрасли. Отмечается, что отношение производительность/стоимость достигнет к концу 10-летия 1 млн. операций/долл. Уменьшатся ограничения при конструировании ИС, что приведет к усложнению связей в ИС. К-МОП-технология будет преобладающей в производстве экономичных ИС. Увеличится выпуск однокристальных микроЭВМ. В процессе конструирования ИС и микроЭВМ будут шире применяться ЭВМ. Рассматриваются различные стратегии конструирования МП.

Отмечаются преимущества иерархического параллельного конструирования. В качестве примера приводятся сведения о проектировании 16-разрядного МП типа Bellmac-32A, который построен на базе одной ИС, содержащей 146000 транзисторов. МП потребляет 0,7 Вт, работает на частоте 8 Мгц и имеет набор команд, позволяющий использовать язык программирования С и операционную систему UNIX. Конструирование и отладка длились 14 месяцев.

УДК 681.325.54

Ю. И. Торгов

ОДНОКРИСТАЛЬНЫЙ КОНТРОЛЛЕР ПРЯМОГО ДОСТУПА К ПАМЯТИ КР580ВТ57

Контроллеры прямого доступа к памяти позволяют существенно улучшить характеристики микроЭВМ при интенсивном обмене информацией между внешними устройствами и ЗУ.

Однокристальный 4-канальный программируемый контроллер прямого доступа к памяти (ПДП) КР580ВТ57 (рис. 1) рассчитан на использование в микроЭВМ на базе микропроцессора КР580ИК80. Применение этого контроллера с микропроцессорами других серий возможно, но требует усложнения схем сопряжения с сигнальными шинами.

Функциональное назначение контроллера

До начала работы контроллера микропроцессор по программе инициализации программирует работу активных каналов: задаются начальный адрес блока данных в ЗУ и количество слов, передаваемых между ЗУ и ВУ, направление передачи данных и особенности режима работы каналов. Затем контроллер обеспечивает автономный (т. е. без микропроцессора) обмен данными.

Инициаторами обмена, определяющими моменты выполнения и темп, являются ВУ. Они выдают на четыре входа контроллера ЗПД0—ЗПД3 активные сигналы «запроса прямого доступа». В ответ на эти сигналы контроллер отключает микропроцессор от сигнальных шин микроЭВМ и подключается к ним в качестве активного устройства управления, определяет наиболее приоритетное ВУ из числа одновременно требующих обслуживания, выдает на управляющие шины сигналы управления, необходимые для передачи информации по шине данных между источником и приемником.

При обмене данными под управлением микропроцессора в системе только один абонент участвует в обмене данными с микропроцессором и генерируется только один системный сигнал — записи или чтения. При обмене данными под управлением контроллера ПДП участвуют два абонента — ячейка ЗУ и внешнее устройство. Поэтому контроллер всегда генерирует пару системных сигналов — чтение и запись.

Другое отличие связано с тем, что контроллер генерирует адрес только ячейки ЗУ, участвующей в обмене. Вместо адреса ВУ контроллер генерирует индивидуальный для каждого канала (0—3) сигнал «разрешение прямого доступа» (РПД). Внешнее устройство должно использовать этот сигнал вместе с сигналом записи или чтения, чтобы принять или выдать код данных на шину данных.

Если сигнал ЗПД поддерживается активным постоянно, до конца пересылки всего блока данных (режим группового обмена), скорость обмена максимальна, но микропроцессор теряет контроль над системой на все время обмена и не реагирует на возникшие в

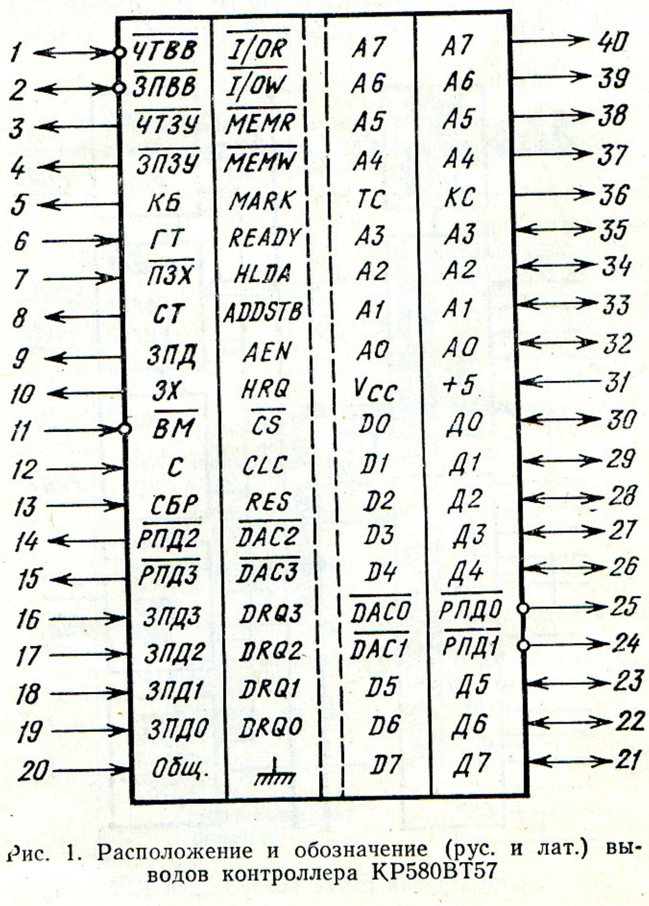


Рис. 1. Расположение и обозначение (рус. и лат.) выводов контроллера КР580ВТ57

это время сигналы прерывания, что не всегда допустимо. В случае снятия сигнала ЗПД со стороны ВУ после каждого цикла обмена, управление периодически передается микропроцессору, а контроллер становится подчиненным устройством. Завершение передачи запрограммированного количества слов (блока данных) в контроллере определяется аппаратно и сопровождается выдачей сигнала «конец счета». Этот сигнал можно использовать для изменения режимов работы аппаратуры, запуска программ, ожидающих завершения процесса обмена в режиме прямого доступа и т. п.

Внутреннее устройство контроллера

На рис. 2 показана структурная схема контроллера ПДП, а на рис. 3 — временна́я диаграмма для его работы. Все четыре канала прямого доступа содержат регистр адреса (РА) и регистр управления (РУ). РА и РУ — 16-разрядные регистры. Чтобы записать через шины данных в регистр 16 разрядов, необходимо выполнить последовательно две команды записи по одному и тому же адресу соответствующего регистра: первая команда записывает данные в регистр младшего байта, а вторая — старшего байта. В РА загружается адрес первой ячейки, с которой должен произойти обмен в ПДП. После каждого цикла обмена содержимое РА увеличивается на 1. Число, загруженное в 14 младших разрядов РУ, определяет число циклов ПДП минус 1, прежде, чем появится сигнал «конец счета». Два старших разряда РУ определяют тип операции, выполняемой каналом. В процессе обмена они не изменяются, но между пересылками блоков данных их можно перепрограммировать. Каждый канал после разрешения на его работу реагирует на входной сигнал ЗПД и выдает сигнал РПД, информирующий ВУ о том, что оно участвует в цикле ПДП. ЗПД0—ЗПД3 — это индивидуальные асинхронные сигналы, вырабатываемые внешним устройством для получения цикла ПДП. Если не включен режим «циклического приоритета», то приоритет сигнала ЗПД0 — высший, а ЗПД3 — низший.

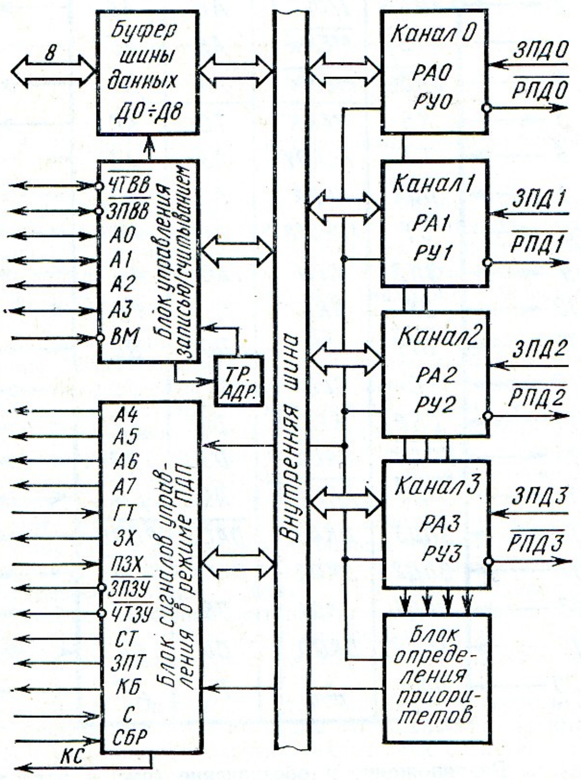


Рис. 2. Структурная схема контроллера КР580ВТ57

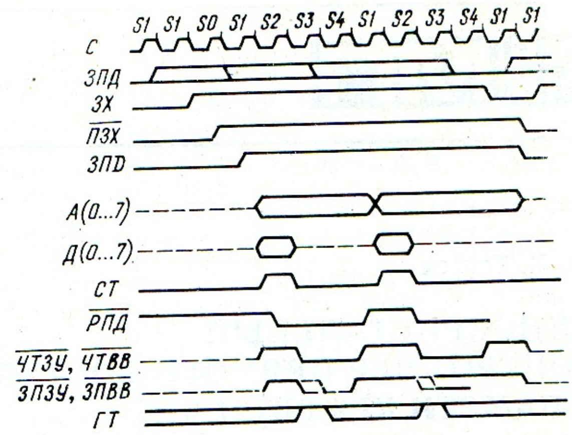


Рис. 3. Временна́я диаграмма работы контроллера КР580ВТ57

Сигнал ЗПД должен поддерживаться активным до появления ответного сигнала РПД, а затем может быть деактивирован, если между циклами ПДП необходимо возвращать управление микропроцессору. Если установлен режим «циклического приоритета», то приоритет каждого канала изменяется в последовательности ...→0→1→2→3→0→... и т. д. после каждого цикла ПДП (а не после каждого запроса ЗПД). Таким образом, только что обслуженный канал получает низший приоритет, а следующий за ним по номеру — высший.

В режиме «фиксированных приоритетов» работа высокоприоритетного ВУ без снятия сигнала ЗПД до конца пересылки блока данных может блокировать работу менее приоритетных ВУ. В режиме «циклического приоритета», и даже группового обмена, это исключено. Отметим, что правило приоритетов ис-

пользуется только при одновременном появлении нескольких сигналов запроса. В остальных случаях, независимо от текущего приоритета канала, он обслужит поступивший запрос сразу.

Связь с шинами микроконтроллера

Контроллер подключается к шине данных через 8-разрядный двунаправленный буфер. Через этот буфер осуществляется доступ к внутренним регистрам контроллера, когда он является подчиненным устройством, и выдается восемь старших разрядов адреса из РА канала, работающего в данном цикле ПДП. Адресная информация выдается в начале цикла ПДП и должна быть записана в буферный регистр адреса по генерируемому контроллером сигналу «старший байт адреса» (СТ), после чего шина данных освобождается для пересылки по ней данных между ЗУ и ВУ.

Выдача старших разрядов адреса ПДП с буферного регистра в системную шину адреса разрешается выходным сигналом контроллера «запрет» (ЗПТ). Он же одновременно должен запретить работу всех устройств системы, которые могут ошибочно использовать системные сигналы в цикле ПДП. Когда контроллер находится в состоянии подчиненного устройства, управление чтением-записью во внутренние регистры контроллера осуществляется с помощью пяти адресных шин А0—А3 и ВМ (выбор микросхемы), сигналов «чтение из ВУ» (ЧТВВ) и «запись в ВУ» (ЗПВВ) (табл. 1).

В режиме ПДП вход ВМ автоматически отключается, чтобы исключить ошибочную выборку контроллера.

Адресные входы А0—А3 становятся выходами, через которые контроллер выдает сигналы управления чтением-записью в устройства, участвующие в цикле ПДП.

Таблица 1

Адресация внутренних регистров

Регистр	ВМ	А3	А2	А1	А0
РА0	0	0	0	0	0
РУ0	0	0	0	0	1
РА1	0	0	0	1	0
РУ1	0	0	0	1	1
РА2	0	0	1	0	0
РУ2	0	0	1	0	1
РА3	0	0	1	1	0
РУ3	0	0	1	1	1
РР (Только запись)	0	1	0	0	0
РО (Только чтение)	0	1	0	0	0
Нет доступа	1	X	X	X	X

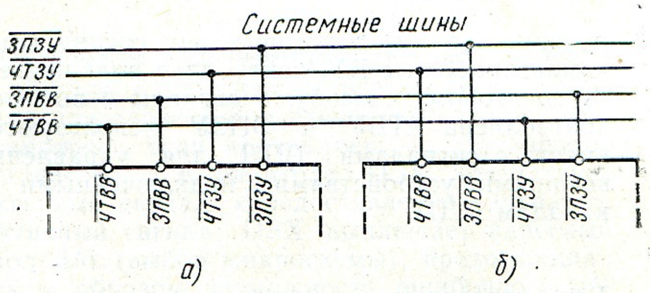


Рис. 4. Способы подключения контроллера КР580ВТ57 к системным шинам

В системе с микропроцессором КР580ИК80 контроллер ПДП можно подключить к системным сигналам запись-считывание двумя способами (рис. 4).

В первом способе (см. рис. 4, *а*) запись или считывание содержимого регистров контроллера осуществляется так же, как запись или считывание ячеек ВУ.

Во втором способе (см. рис. 4, *б*) для этого можно использовать более широкий набор команд работы с памятью, что позволяет составлять короткие программы. В зависимости от способа подключения, изменяется смысл кодирования режима работы каналов, устанавливаемый двумя старшими разрядами РУ (табл. 2). При работе контроллера с мик-

Таблица 2

Режимы работы каналов контроллера ПДП в зависимости от способа подключения к системным сигнальным шинам

Разряды $РУ_{15}$	Разряды $РУ_{14}$	Операция	Вырабатываемые контроллером системные сигналы: записи	Вырабатываемые контроллером системные сигналы: чтения
0	0	Верификация в цикле ПДП	Нет	Нет
0	1	Чтение из ВУ-запись в ЗУ	$\overline{\text{ЧТВВ}}$	$\overline{\text{ЗПЗУ}}$
1	0	Чтение из ЗУ—запись в ВУ	$\overline{\text{ЧТЗУ}}$	$\overline{\text{ЗПВВ}}$
1	1	Неразрешенная комбинация	—	—
0	0	Верификация в цикле ПДП	Нет	Нет
0	1	Чтение из ЗУ-запись в ВУ	$\overline{\text{ЧТЗУ}}$	$\overline{\text{ЗПВВ}}$
1	0	Чтение из ВУ-запись в ЗУ	$\overline{\text{ЧТВУ}}$	$\overline{\text{ЗПЗУ}}$
1	1	Неразрешенная комбинация	—	—

ропроцессорами, у которых нет специальных команд работы с ВУ, контроллер подключается по второму способу, а сигналы с выходов контроллера ЧТВУ и ЗПЗУ используются вместе с сигналами РПД для управления внешними устройствами, подключенными к каналам ПДП.

Блок управления

В блок управления входят 8-разрядные регистр режимов и регистр состояния, а также логические схемы, вырабатывающие сигналы и обеспечивающие работу системы в цикле ПДП.

*Сигналы**

Сброс (СБР)	(входной)	— Устанавливает в нуль содержимое внутренних регистров, внутренний триггер адреса, запрещает работу каналов ПДП.
Синхросигнал (С)	»	— Тактовый сигнал, на основе которого формируется внутренняя временная диаграмма работы контроллера (T_c = =0,5 мкс).
Готовность (ГТ)	»	— При отсутствии сигнала ГТ удлиняется цикл ПДП для согласования временной диаграммы медленно работающих устройств. Сигнал ГТ стробируется в начале внутреннего такта контроллера.
Представление захвата (ПЗХ)	»	— Сигнал, переводящий контроллер в режим главного устройства; далее контроллер осуществляет цикл ПДП.
Захват шин (ЗХ)	(выходной)	— Сигнал выдается контроллером на вход процессора, после чего процессор отключается от системных шин, извещая об этом контроллер сигналом ПЗХ. Сигнал ЗХ появляется при наличии сигнала ЗПД на входе активированного канала.
Адрес ША4-ША7	»	— В цикле ПДП выдаются разряды РА4-РА7 регистра адреса канала. В остальное время — выключены (высокое сопротивление).
Запись в ЗУ (ЗПЗУ)	(выходной)	— Сигнал записи } вырабатываются контроллером в цикле ПДП.
Чтение из ЗУ (ЧТЗУ)	»	— Сигнал чтения } вырабатываются контроллером в цикле ПДП.
		Использование сигналов определяется способом подключения контроллера к сигнальным шинам системы.
Старший байт адреса (СТ)	»	— В цикле ПДП разрешается запись старшего байта адреса ПДП с шины данных во внешний буферный регистр адреса.
Запрет	»	— В цикле ПДП разрешает выдачу адреса с внешнего буферного регистра на системную шину адреса (старший байт) и запрещает работу устройств, не участвующих в ПДП.
Конец счета (КС)	»	— Выдается в последнем цикле передачи блока данных (при содержимом РУ=0); запрещает дальнейшую работу канала, если установлен разряд РР6.
Конец блока (КБ)	»	— Сигнал указывает, что до конца передаваемого блока необходимо выполнить число циклов, кратное 128.

* Алексенко А. Г., Галицын А. А., Иванников А. Д. Проектирование радиоэлектронной аппаратуры на микропроцессорах. — М.: Радио и связь, 1984.

Регистр режима (РР). Каждый разряд регистра режима имеет самостоятельное значение, которое записывается в регистр после того, как были запрограммированы РА и РУ соответствующего канала.

Разряды РР0—РР3 в состоянии 1 разрешают, а в состоянии 0 — запрещают работу каналов с номером, равным номеру разряда.

Разряд РР4 в состоянии 0 устанавливает фиксированные приоритеты каналов, а в состоянии 1 — циклические.

Разряд РР5 в состоянии 0 обеспечивает генерацию сигналов записи по обычной временно́й диаграмме, а в состоянии 1 — на один такт синхросерии раньше. Это позволяет устройствам, генерирующим сигнал готовности по фронту сигнала записи, вырабатывать сигнал ГТ раньше и избегать перехода контроллера в состояние ожидания и связанных с этим задержек.

Разряд РР6 в состоянии 1 обеспечивает сброс соответствующего разряда РР0—РР3 разрешающего работу канала, в цикле ПДП которого был выработан сигнал КС. При этом канал выключается и дальнейшие запросы ЗПД на него не оказывают влияния, пока канал не будет повторно перепрограммирован. Если РР6 установлен в состояние 0, канал будет обычным образом реагировать на сигналы ЗПД и после появления сигнала КС, но информация в регистрах РА и РУ может оказаться недействительной. Поэтому ВУ должно само прекратить выработку сигналов

ЗПД, чтобы предотвратить возможность появления ошибок.

Разряд РР7 (разряд автозагрузки) в состоянии 1 обеспечивает возможность использования повторяющегося аппаратного перепрограммирования канала 2 по сигналу КС. Это сокращает потери времени на перепрограммирование канала по сравнению с программным методом. При этом РР6=1 не останавливает работу канала 2. Параметры, загружаемые в РР2 и РУ2, аппаратно пересылаются из регистров канала 3 (РР3 и РУ3) без изменения их содержимого.

Когда РР7=1, программно загружаются каналы 2 и 3 (одновременно). Это позволяет программировать только один канал, если требуется обеспечить повторяющиеся пересылки блоков данных с теми же параметрами. Примером может служить работа с дисками или дисплейным контроллером.

Если необходимо организовать последовательность пересылок нескольких блоков с разными параметрами, следует параметры первого блока загрузить в канал 2, а параметры второго и последующих — в канал 3. Перепрограммировать канал 3 можно во время работы канала 2, за исключением моментов автоматической перегрузки параметров в канал 2. Каждый раз, когда после сигнала КС в контроллере происходит автозагрузка, в регистре состояния устанавливается в 1 разряд РС4. Программа загрузки параметров в канал 3 должна опрашивать РС4, чтобы исключить возможность обращения к регистрам канала 3 в это время.

Параметры в канал 2 пересылаются в начале первого после КС цикла ПДП канала 2. В конце этого цикла разряд РС4 сбрасывается в «0», после чего программно можно загружать новые параметры в регистры канала 3.

Следует отметить, что если работа канала 3 разрешена, то он и в режиме автозагрузки будет работать как обычно, но параметры в его регистрах будут меняться после каждого цикла ПДП и в измененном виде записываться в регистры канала 2.

Регистр состояния РС.

Разряды РС0—РС3 устанавливаются в «1», когда при работе канала с номером, равным номеру разряда, возникает сигнал КС. Эти разряды остаются в единичном состоянии до считывания содержимого РС, после чего сбрасываются в «0». Сигнал СБР сбрасывает в «0» все разряды РС.

Разряд РС4 устанавливается и сбрасывается при описанных выше условиях автозагрузки. Считывание РС на него не влияет, но его можно сбросить, установив в «0» разряд автозагрузки РР7.

Особенности программирования контроллера

Доступ к внутренним регистрам контроллера для записи или чтения их содержимого возможен, если контроллер — подчиненное устройство, т. е. на входе ПЗХ отсутствует активный сигнал «предоставление захвата». Активный сигнал ПЗХ выключает адресный вход ВМ (выбор микросхемы), предотвращая таким образом возможность ошибочного обращения к контроллеру при выполнении циклов обмена в режиме прямого доступа.

Каждый канал контроллера имеет 16-разрядные регистр адреса и регистр управления и, кроме того, 8-разрядные общие для каналов регистр состояния и регистр режима. Так как обращение к регистрам идет через 8-разрядную шину данных, используется внутренний триггер адреса, состояние которого и определяет доступ к старшей или младшей половине 16-разрядных регистров. Этот триггер адреса устанавливается в исходное состояние по сигналу «сброс», а также при каждой записи кода в РР. При этом обеспечивается доступ к младшему байту канальных регистров. Триггер адреса переключается после каждого обращения в адрес канального регистра. Поэтому необходимо организовывать парные обращения к канальному регистру, чтобы получить доступ сначала к младшему, а затем к старшему байту. Недопустимо переключение сигнала $\overline{ВМ}$ во время активного сигнала на входах $\overline{ЗПВВ}$ или $\overline{ЧТВВ}$, так как это вызовет ошибочное переключение триггера адреса. По этой же причине нельзя разделять парные циклы обращений к канальным регистрам.

В микропроцессорной системе, работающей с прерываниями, следует запрещать прерывания перед выполнением двойного обращения к регистрам. Такая ситуация, в частности, может возникнуть при программировании каналов ПДП в результате реакции на прерывание по какому-либо событию. Программирование канала сводится к записи 16-разрядного начального адреса обмена в РА (парное обращение), программированию режима работы (старшие два разряда РУ) и длины передаваемого блока данных (младшие 14 разрядов РУ), затем записывается 8-разрядный код в РР, разрешающий работу запрограммированных каналов. После сигнала «сброс» РР содержит все нулевые разряды, запрещая таким образом работу всех каналов контроллера. Нельзя разрешать работу канала, если предварительно в канальные регистры не были записаны действительные значения параметров, так как случайный сигнал запроса от внешнего устройства может вызвать цикл ПДП и внести ошибки в содержимое ЗУ.

Организация передачи массивов с непоследовательными адресами и расширение разрядности адреса с помощью контроллера ПДП

Часто необходимо передать массивы данных, занимающие в ОЗУ область с непоследовательными адресами. Так, в задачах обработки изображений, сигналов и т. п. приходится обрабатывать элементы, расположенные по заданному закону вокруг точки с меняющимися координатами.

Вычисление адресов элементов требует существенных затрат микропроцессорного времени при программном режиме обмена. Чтобы использовать контроллер ПДП, вырабатывающий строго последовательные адреса при пересылке массива, можно применить двухступенную выработку адреса с помощью комбинации из табличного ЗУ и параллельного сумматора.

В табличном ЗУ могут храниться либо поправки (ΔA) к базовому адресу (А ПДП), вырабатываемому контроллером при очередном цикле обращения к элементу массива, либо готовый исполнительный адрес ($A_{\text{исп. г}}$) (рис. 5).

В первом случае требуется меньшая разрядность табличного ЗУ, но наличие параллельного сумматора и счетчика циклов обращения к ЗУ существенно усложняет схему. Во втором необходимо полноразрядное табличное ЗУ, но упрощается схема устройства. В частности, можно использовать адрес, вырабатываемый контроллером, как адрес обращения к табличному ЗУ.

Табличное ЗУ организуется как двухвходовое. Через первый вход оно загружается с помощью микропроцессора до начала работы контроллера. Через второй вход — к табличному ЗУ обращается контроллер.

Аналогично можно расширить разрядность адреса, вырабатываемого при обращении к ЗУ большого объема, используя табличное ЗУ в качестве набора регистров «приписки». В формирование адреса выбора регистра используется сигнал ППД, определяющий номер активного канала в данном цикле обращения, а также часть разрядов адреса, выдаваемого контроллером.

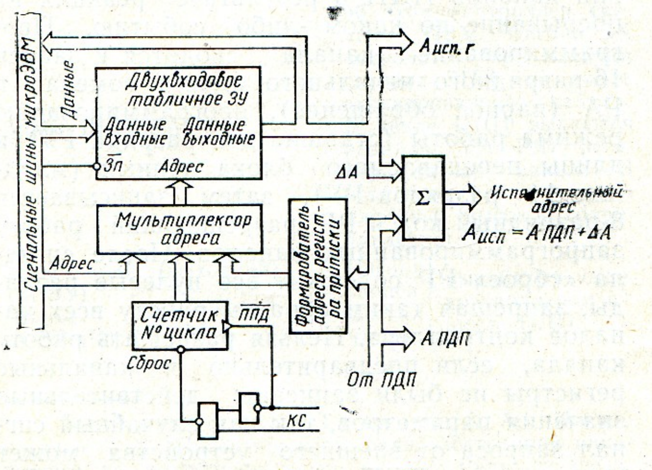


Рис. 5. Использование табличного ЗУ при двухступенном формировании исполнительного адреса в ПДП

Повышение скорости обмена с ЗУ

В режиме блочного обмена, когда сигнал ЗПД не снимается до конца передачи всего массива данных, выдаваемые контроллером адреса меняются с периодом в четыре тактовых импульса. Длительность фронтов на адресных шинах при переключении кодов составляет менее 100 нс. Используя внешний синхронный счетчик-делитель частоты и схему формирования временно́й диаграммы сигналов обращения к ЗУ, можно обеспечить цикл ЗУ примерно 400 нс. Для устойчивой работы контроллера при этом требуется обеспечить фазовую синхронизацию сигналов ГТ, ЗПД и ПЗХ с помощью цепочек на логических элементах (рис. 6).

Использование контроллера КР580ВТ577 с микроЭВМ «Электроника 60»

В соответствии с протоколом запроса и представления сигнальных шин, принятым в микроЭВМ, «Электроника 60» (рис. 7) выдает в канал сигнала запроса прямого доступа (К—ТПД—Н) при появлении сигналов ЗХ на выходе любого из двух контроллеров. При этом подготавливаются к установке триггера подтверждения выбора (ТР.ВБ) и прямого доступа к памяти (ТР.ПДП). Триггеры устанавливаются по сигналу предоставления пря-

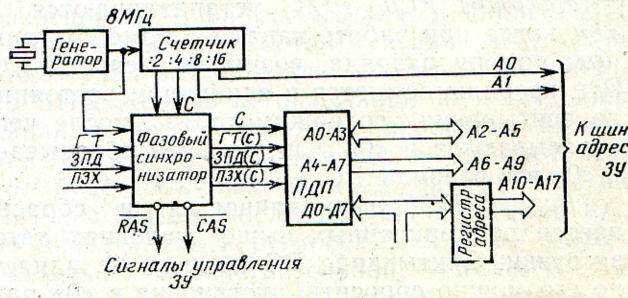


Рис. 6. Организация укороченных циклов

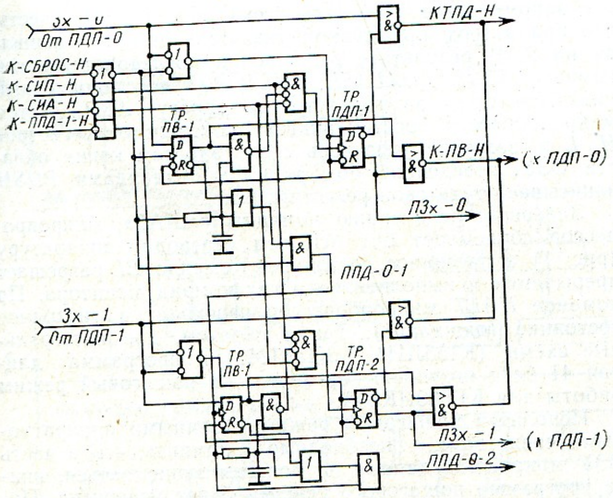


Рис. 7. Логическая схема подключения сигналов запроса и предоставления прямого доступа от двух контроллеров КР580ВТ57 к каналу микроЭВМ «Электроника 60»

мого доступа (К—ППД—Н), который вырабатывается процессором после освобождения канала предыдущим активным устройством.

Установка ТР.ПДП прекращает выработку сигнала К—ТПД—Н, выдает сигнал К—ПВ—Н, извещающего процессор о занятости канала и сигнал ПЗХ, разрешающий работу контроллера ПДП в активном режиме. Гирляндная цепочка, по которой распространяются сигналы ППД-1 — ППД-0, организует приоритетное включение контроллеров при одновременном появлении запросов на обслуживание от внешних устройств, подключенных к разным контроллерам. При подключении добавочных контроллеров для организации их приоритетного обслуживания используется сигнал ППД-0-2.

Статья поступила 3 августа 1984 г.

УДК 681.3.06

А. В. Найденов, А. А. Туманов, В. А. Романенков

ПОШАГОВЫЙ РЕЖИМ ПРИ РАЗРАБОТКЕ И ОТЛАДКЕ ПРОГРАММ ДЛЯ МИКРОПРОЦЕССОРНЫХ СРЕДСТВ НА БАЗЕ БИС КР580ИК80А

Две интегральные схемы (К155ЛИ1 и К155ТМ2) и программа длиной 41 байт позволяют организовать пошаговый режим работы для БИС КР580ИК80А.

При построении систем управления на базе одноплатных микроЭВМ или микроконтроллеров один из важнейших и самых трудоемких этапов в разработке всей системы — создание и отладка программ.

Разрабатывать программы (объемом 1—2 Кбайт) в кодах можно, используя входящую, как правило, в указанные микропроцессорные устройства специальную программу — управляющий монитор (1—2 Кбайт). 10—15 команд монитора позволяют оператору работать с регистрами микропроцессора, ячейками памяти ОЗУ, портами ввода-вывода микроЭВМ; управлять выполнением программ. Команды монитора выполняются как правило, в режиме прерывания микропроцессора в составе микроЭВМ или микроконтроллера.

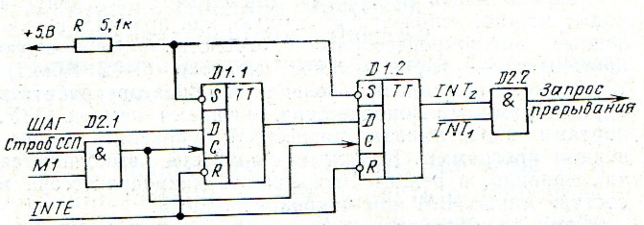
Один из эффективных способов отладки программы — ее выполнение в пошаговом режиме, т. е. при каждом повторном пуске программы выполняется только одна команда. Микропроцессор останавливается, а на индикатор микроЭВМ выводится определенная информация (содержимое регистров, ячеек памяти, слово состояния микропроцессора). Для перехода к этому режиму используется команда монитора «ШАГ». Выполнить команду «ШАГ» можно благодаря особенностям микропроцессора КР580ИК80А: цикл выборки любой команды сопровождается в слове состояния процессора высоким уровнем сигнала М1; запрос прерывания (в случае его разрешения), воспринимается микропроцессором в последнем такте выполняемой команды; при разрешении прерывания, т. е. выполнении команды *EI*, на выводе микропроцессора *INTE* появляется высокий уровень сигнала (при запрещении — низкий).

Для работы в пошаговом режиме оператор при помощи соответствующей команды монитора заносит в программный счетчик микропроцессора исходный адрес и нажимает клавишу «ШАГ». Микропроцессор выполняет мониторную подпрограмму «ПУСК»: настраивает аппаратуру схемы пошагового режима (см. рисунок), восстанавливает содержимое регистров для программы пользователя из фиксированной в мониторе области ОЗУ, — выполняет текущую команду проверяемой программы пользователя и мониторную подпрограмму «СОХРАНЕНИЕ»: записывает в фиксированную область ОЗУ новое содержимое регистров микропроцессора, выключает аппаратуру пошагового режима, — после чего по команде *HALT* аналогично выполняется следующая команда программы пользователя.

Рассмотрим более подробно взаимодействие аппаратуры и программного обеспечения. Исходное состояние микропроцессора — состояние останова (*HALT*) и прерывания в нем разрешены (после выполнения команда EI). При нажатии оператором клавиши «ШАГ» первый запрос прерывания (*INT1*) выводит микропроцессор из состояния ожидания. Принимая вектор прерывания, сопровождающий этот запрос (или команду перехода на подпрограмму, если в систему входит блок КР580ВН59), микропроцессор переходит к подпрограмме «ПУСК»:

MVIA, 1	— запуск аппаратной схемы;
OUT STEP	— установка бита М1;
LXI SP, RA	— установка указателя на стык хранения регистров;
POP B	— восстановление регистров *B*, *C*;
POP D	— восстановление регистров *D*, *E*;
POP H	— восстановление регистров *H*, *L*;
LXI SP, ASP	— восстановление указателя стека;
EI	— разрешение прерывания;
JMP	— переход на текущий адрес программы;

START:<PC>
RA:<BC>
<DE>
<HL>
RA1:<AF>
ASP:<SP>
— область сохранения регистров микропроцессора



Аппаратура режима: строб *ССП* — строб слова состояния процессора; *M1* — один из битов ССП; *INT* — входной сигнал микропроцессора о запросе прерывания; *INTE* — выходной сигнал микропроцессора о разрешении прерывания

Вначале микропроцессор устанавливает бит шагового режима в регистре *STEP*, и на информационный вход триггера *D1.1* (рис. 1) поступает сигнал высокого уровня. Затем командой *LXI SP, RA* микропроцессор устанавливает указатель стека на начало фиксированной области ОЗУ, где хранится содержимое регистров программы пользователя, которые далее восстанавливаются командами *POP*... Для восстановления исходного состояния указателя стека микропроцессор использует команду *LXI SP, ASP*. Последующей командой микропроцессор выполняет сразу две функции: разрешает прерывания и, кроме того, сигнал высокого уровня, появляющийся на выходе *INTE* микропроцессора, разрешает работу триггеров *D*1.1, *D*1.2, *D*2.1, *D*2.2 (см. рисунок).

Особенность подпрограммы «ПУСК» — это выполнение команды *JMP* <содержимое программного счетчика>. По этой команде программный счетчик микропроцессора одновременно восстанавливается и запускается для выполнения новой команды. В ОЗУ содержимое программного счетчика хранится сразу за первым байтом команды *JMP*, т. е. является ее аргументом. Команда *JMP* сопровождается (при выборке из ОЗУ) битом *M1* в слове состояния процессора, в результате чего на выходе триггера *D*1.1 появляется высокий уровень сигнала. Когда микропроцессор выполняет текущую команду проверяемой программы, на триггеры поступает еще один бит M1. Высокий уровень сигнала на выходе триггера *D*1.2 вызывает второе прерывание микропроцессора. По этому прерыванию выполняется подпрограмма «СОХРАНЕНИЕ»:

XTHL — обмен между содержимым регистров *H, L* и верхней парой ячеек стека;
SHLD START — операция сохранения программного счетчика по адресу *START*;
LX1, H,2 — учет верхнего слова в стеке;
DAD SP — запись в *HL* значения значения *SP* перед прерыванием;
SHLD ASP — операция сохранения действительного значения указателя стека по адресу *ASP*;
XTHL — восстановление содержимого *H, L*;
LX1 SP, RA1 — загрузка в указатель стека адреса стека хранения регистров;
PUSH PSW — сохранение аккумулятора (A) и флагов (F);
PUSH H — сохранение регистров *H, L*;
PUSH D — сохранение регистров *D, E*;
PUSH B — сохранение регистров *B, C*;
MVIA O — обнуление аккумулятора;
OUT STEP — выключение аппаратной схемы;
EI — разрешается прерывание;
HALT — останов.

После выполнения команды *XTHL* в регистрах *H, L* оказывается новое содержимое программного счетчика, занесенное в стек при втором прерывании. Командой *SHLD START* микропроцессор заносит его в ячейку, отведенную для хранения программного счетчика. Далее запоминается указатель стека. Необходимо учесть, что при втором прерывании указатель стека уменьшился на 2. В результате выполнения последовательности команд *LXI H,2*; *DAD SP*; *SHLD ASP* восстановленный указатель стека записывается по адресу *ASP*. Затем микропроцессор восстанавливает содержимое регистров *H, L* и, загрузив указатель стека адресом конца области ОЗУ, отведенной под регистры, командами *PUSH* записывает туда свои регистры.

Завершая реализацию команды «ШАГ», микропроцессор сбрасывает бит M1, чем переводит аппаратуру (рис. 1) в пассивное состояние. Команда *EI* разрешает прерывания и выполнение других команд монитора. По команде *HALT* микропроцессор переходит в исходное состояние (ожидание). Таким образом, две интегральные схемы (К155ЛИ1 и К155ТМ2) и программа длиной 41 байт позволяют организовать пошаговый режим работы для КР580ИК80А.

Используя пошаговый режим с данными аппаратурными средствами, необходимо помнить, что в часть ОЗУ микроЭВМ данные не должны записываться, иначе программа пошагового режима будет испорчена. Поэтому для надежной работы команды монитора «ШАГ» необходима аппаратная защита программ «ПУСК» и «СОХРАНЕНИЕ».

Статья поступила 5 марта 1984 г.

УДК 681.326—181.48

Н. В. Ширшиков

ДВУХУРОВНЕВАЯ ОРГАНИЗАЦИЯ МИКРОМАШИННОГО КОМПЛЕКСА

Комплекс из двух микроЭВМ «Электроника 60» эффективно использует адресное пространство микроЭВМ и гарантирует надежность, необходимую для непрерывной 2—3-недельной работы.

Двухуровневая архитектура микромашинного комплекса продиктована необходимостью применения перепрограммируемых постоянных запоминающих устройствах (ППЗУ) для хранения системных и прикладных (на языке Бейсик) программ и позволяет удвоить адресное пространство комплекса в целом, чтобы обеспечить функционирование прикладных программ с объемом 56 Кбайт.

Микромашинный комплекс был разработан в Научно-исследовательском вычислительном центре АН СССР для исследований в области биотехнологии [1]. Микромашинный комплекс реализован из двух функционально связанных универсальных микроЭВМ «Электроника 60» [2]. Одна из них (верхний уровень) — управляющая микроЭВМ, другая (нижний) — программируемый контроллер (рис. 1). Машины комплекса связаны через устройства последовательного обмена (УПО) [3, 4].

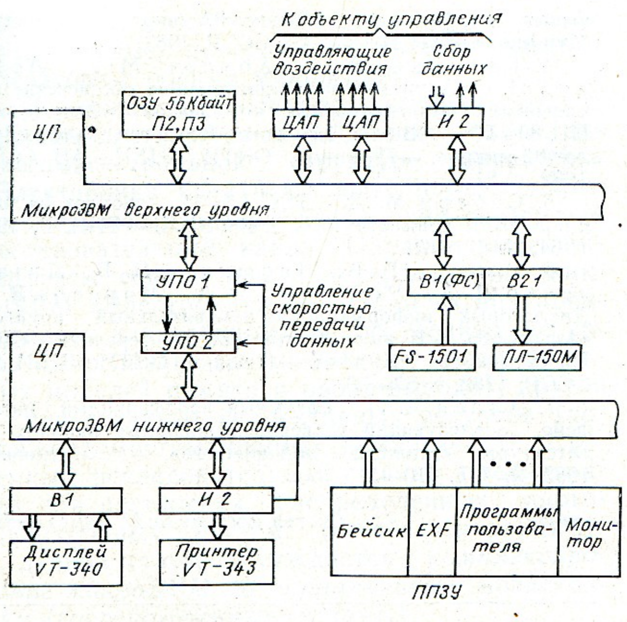


Рис. 1. Архитектура двухуровневого комплекса

МикроЭВМ верхнего уровня обрабатывает данные и управляет микробиологическим экспериментом.

МикроЭВМ нижнего уровня — программируемый терминал для микроЭВМ верхнего уровня. Регистры устройства последовательного обмена управляющей микроЭВМ (верхний уровень) имеют адреса пульта ASCII, т.е. УПО1 — аналог модуля В1 микроЭВМ «Электроника 60». МикроЭВМ нижнего уровня, связанная с УПО1 устройством УПО2, исполняет под управлением подпрограммы монитора (о нем будет сказано ниже) функции пультового терминала. Благодаря этому программируемый контроллер (микроЭВМ нижнего уровня), инициирует загрузку и пуск программ управляющей микроЭВМ. Для инициализации используется записанная изготовителем в БИС ЦП микроЭВМ «Электроника 60» микропрограмма обслуживания пульта ASII [5].

Адресное пространство микроЭВМ нижнего уровня практически полностью используется в качестве ППЗУ. В ППЗУ хранятся абсолютные модули системы Бейсик, библиотека ассемблерных программ (внешних функций), а также тексты прикладных программ на языке Бейсик. В качестве резидентной программной системы используется интерпретатор Бейсик (8 842.006—20), расширенный подпрограммами реального времени на языке ассемблер. Модуль интерпретатора Бейсик и связанные с ним модули ассемблерных подпрограмм загружаются в управляющую микроЭВМ нижнего уровня. Вопросы прикладного программирования и функционирования программ реального времени изложены в работе [6].

УПО выбрано для связи микромашин в комплексе потому, что по реализации программно доступных регистров, битам готовности и разрешения прерывания модуль УПО полностью представим в качестве контроллера (интерфейса) пультового терминала — платы В1.

Для данной реализации межмашинной связи существенна также возможность управления скоростью передачи УПО. Это необходимо из-за неравнозначности временных характеристик различных внешних устройств при операциях ввода-вывода, затребованных от управляющей микроЭВМ [7]. Скорость передачи данных 9600 бод используется при обращении к ППЗУ и загрузке программ в управляющую микроЭВМ, а 300 бод — для работы с внешними устройствами типа VT-340, VT-343, функционирующими с микроЭВМ нижнего уровня. Скорости передачи данных переключаются программно. Скоростями передачи данных УПО1 и УПО2 можно управлять аппаратно: подачей ТТЛ-совместимых потенциальных сигналов от шин PC00 PC01 платы И2 на контакты перемычек установки скоростей передачи данных УПО1 и УПО2.

Монитор (рис. 2) программно управляет взаимодействием двух микроЭВМ в комплексе. Он размещается в ППЗУ микроЭВМ нижнего уровня и выполняет:

— инициализацию и загрузку Бейсика с внешними функциями в микроЭВМ верхнего уровня;

— эмуляцию пультового терминала на нижнем уровне комплекса;

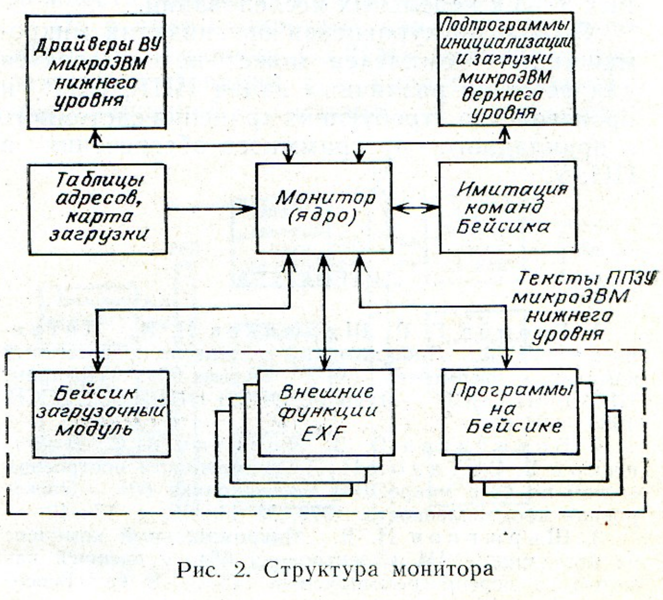


Рис. 2. Структура монитора

— организацию доступа управляющей микроЭВМ к периферийным устройствам, аппаратно связанным с микроЭВМ нижнего уровня;

— управление загрузкой и стиранием прикладных программ.

Монитор интерпретирует следующие команды для исполнения перечисленных ниже функций:

/JAMP BAS — предоставление пользователю ресурсов Бейсика для программирования;

/LOAD Pi — загрузка i-й программы пользователя;

/DEL Pi — стирание i-й программы пользователя;

/<CTRL>F — передача управления монитору.

К разработке и отладке прикладных программ пользователь переходит командой /JAMP BAS. Отлаженной программе пользователя присваивается номер с соответствующими указателями области ППЗУ, отведенной для хранения данной программы. Программа записывается в ППЗУ на стендовой микроЭВМ «Электроника 60». Число программ пользователя в ППЗУ ограничено лишь адресным пространством микроЭВМ нижнего уровня, позволяющим адресовать не более 56 Кбайт ППЗУ.

Поиск программ пользователя инициируется командой LOAD Pi. Монитор, обнаружив в таблице указателей соответствие номеру програмы, загружает выбранную программу в управляющую микроЭВМ.

Эксплуатация двухуровневого комплекса в Секторе управляемого культивирования ИБФМ АН СССР в 1982—1983 гг. показала, что его надежность достаточна для непрерывных 2—3-х недельных исследований.

Вывод: двухуровневая организация микромашинного комплекса может использоваться для решения различных задач НИР, ОКР и производства, требующих хранения системного и прикладного программного обеспечения в ППЗУ.

ЛИТЕРАТУРА

1. Громов Г. Р., Ширшиков Н. В., Литвиненко Л. А. Лабораторный аппаратно-программный комплекс ферментер—ЭВМ «Альфа-60»: Препринт НИВЦ АН СССР. — Пущино: ОНТИ НЦБИ АН СССР, 1983. — 21 с.

2. Борженков В. Д., Лопатин В. С., Плотников В. В., Талов И. Л. Возможности построения систем на базе микроЭВМ «Электроника 60». — Электронная промышленность, 1979, № 11—12, с. 17—19.

3. Ширшиков Н. В. Микромашинный комплекс на базе микроЭВМ «Электроника 60» с внешней памятью на перепрограммируемых ПЗУ. — В кн.: Всесоюзная конференция «Диалог-82-микро». Тез. докл. — Пущино: ОНТИ НЦБИ АН СССР, 1983, с. 53.

4. Глухман В. Л., Хохлов М. М., Лозовой И. О. Архитектура и программное обеспечение диалоговых вычислительных комплексов «Электроника НЦ-80—20». — В кн.: Тез. докл. Всесоюзн. конф. «Диалог-82-микро». — Пущино: ОНТИ НЦБИ АН СССР, 1982, с. 21.

5. Себерн М. Д. ДЕС LSI-11 — система на базе микроЭВМ, совместимая с мини-ЭВМ. — ТИИЭР, 1976, т. 64, № 6, с. 67.

6. Громов Г. Р., Ширшиков Н. В., Литвиненко Л. А., Ройтберг М. А., Макеева Е. В. Диалоговый информационно-измерительный комплекс «Альфа-60». — В кн.: Всесоюзная конференция «Диалог-82-микро». Тез. докл. — Пущино: ОНТИ НЦБИ АН СССР, 1982, с. 52—53.

7. Аллен А. Л. Разумный периферийный интерфейс, реализующий усовершенствованный вариант архитектуры «главный — подчиненные». — Электроника, 1982, № 7, с. 110—115.

Статья поступила 27 марта 1984 г.

УДК 681.3—181.4.004

В. В. Сумин, А. Б. Васильев

ДВУХМАШИННЫЙ КОМПЛЕКС НА ОСНОВЕ микроЭВМ «ЭЛЕКТРОНИКА 60» И «ЭЛЕКТРОНИКА С5-21М»

Двухмашинный комплекс, объединяющий микроЭВМ «Электроника 60», «Электроника С5-21М» и ряд функциональных модулей условно назван исследовательской двухпроцессорной системой (рис. 1). Организация такой системы не требует аппаратной доработки микроЭВМ: все соединения выполняются по схеме «шина в шину». При отладке систе-

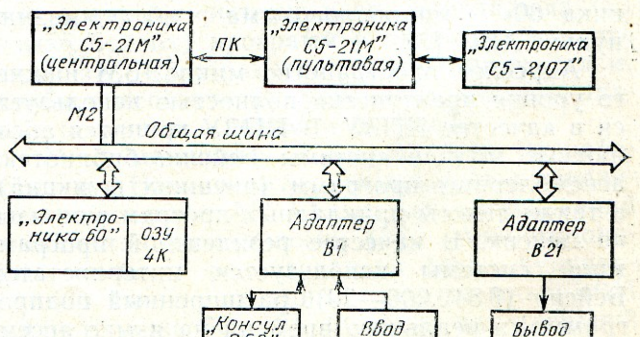


Рис. 1. Структурная схема исследовательской двухпроцессорной системы

мы можно использовать только встроенные (резидентные) программные средства обмена оператора с обеими микроЭВМ.

Возможность создания многомашинной системы достигается прежде всего благодаря **многомагистральной структуре микроЭВМ «Электроника С5-21М»**: два параллельных магистральных интерфейса и один системный последовательный канал. Параллельные интерфейсы совместимы с интерфейсами микроЭВМ «Электроника 60». Один из них предназначен для расширения собственных ресурсов микроЭВМ (подключения дополнительных модулей памяти и ввода-вывода), другой — для создания коллективных ресурсов (построения многомашинных комплексов)*. Системный последовательный канал предназначен для построения распределенных многопроцессорных систем (сетей микроЭВМ). В данной статье обсуждается *комплексирование микроЭВМ по параллельным магистралям* (сосредоточенные системы).

Обмен микроЭВМ комплекса с общей памятью осуществляется в режиме прямого доступа (ПДП). Управление обменом реализуется тремя сигналами: требование ПДП, подтверждение (разрешение) прямого доступа и подтверждение выборки в режиме ПДП.

Пульт оператора системы выполнен на микроЭВМ «Электроника 60», дополненной соответствующими адаптерами и периферийными устройствами. Центральная микроЭВМ «Электроника С5-21М» соединена последовательным каналом с пультовой микроЭВМ, которая осуществляет подготовку и отладку программ модуля обмена «Электроника С5-2107» (см. рис. 1).

Для практической реализации двумашинного комплекса следует проделать обычные монтажные процедуры:

— организовать пультовый режим — связь модуля «Электроника С5-2107» с пультовой микроЭВМ «Электроника С5-21М»;

— выполнить необходимые монтажные соединения для связи центральной и пультовой микроЭВМ «Электроника С5-21М» по последовательному каналу;

— сделать монтаж соединения по схеме «шина в шину» магистрали центральной микроЭВМ «Электроника С5-21М» и микроЭВМ «Электроника 60»;

— адаптеры В1 и А21 микроЭВМ «Электроника 60» соединить с внешними устройствами ПЛ-150М, FS-1501, «Консул 260».

На реализованной исследовательской двухпроцессорной системе проведена серия экспериментов по отработке тестовых программ периферийных устройств (ввод массива с перфоленты, дублирование перфолент, печать с клавиатуры), программ сравнения производительности, организации режимов прерываний и обмена.

Область применения комплексированной системы определяется особенностями и возможностями каждой микроЭВМ комплекса.

МикроЭВМ «Электроника С5-21М» имеет систему команд, обеспечивающую реализацию контроллерных функций. Функциональные модули этой машины ориентированы на связь с объектом управления. МикроЭВМ и модули имеют достаточную конструктивную жесткость, высокую надежность и, следовательно, хорошие возможности для встраивания в машины, оборудование, приборы.

МикроЭВМ «Электроника 60» имеет мощную систему команд, аппаратно реализованные режимы работы с плавающей запятой, богатый фонд программных средств.

В комплексированных системах микроЭВМ «Электроника С5-21М» используется для автоматизации контроллерных функций, а микроЭВМ «Электроника 60» — для целевой обработки данных.

В большинстве случаев применения двухмашинного комплекса — для организации распределенных и сосредоточенных информационно-управляющих систем с программируемыми устройствами связи с объектом — роль ведущей микроЭВМ в комплексе принадлежит микроЭВМ «Электроника 60». МикроЭВМ «Электроника С5-21М» играет роль ведомой и реализует функции локальных контроллеров и регуляторов. По сути дела речь идет о микропроцессорных устройствах связи с объектом, расширяющих функциональные возможности микроЭВМ «Электроника 60».

Но в ряде конкретных применений микроЭВМ «Электроника С5-21М» выполняет диспетчерские функции, т. е. является ведущей в многомашинной системе, а микроЭВМ

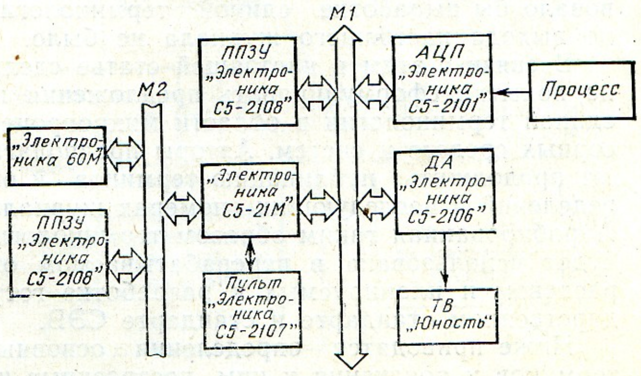


Рис. 2. Структурная схема спектроанализатора

* Гальперин М. П. Одноплатные микроЭВМ и микроконтроллеры. — Микропроцессорные средства и системы, 1984, № 2, с. 16—19.

«Электроника 60» используется в качестве арифметического расширителя, например в анализаторе спектра сигналов в инфракрасном диапазоне (рис. 2).

МикроЭВМ «Электроника С5-21М» с помощью АЦП принимает выборки процесса, масштабирует их, располагает в нужном порядке и пересылает в ОЗУ микроЭВМ «Электроника 60», которая осуществляет быстрое преобразование Фурье, интегрирование спектра и т. д. На экране спектроанализатора отображаются одновременно сам процесс и его спектр. Масштаб по оси ординат для спектра и оцифровка осей координат могут изменяться по требованию оператора. Маркер под управлением оператора указывает нужную частоту и на экране отображается ее значение. На экране в виде ряда указываются также максимумы спектра, интеграл от спектра. Кроме того, путем программирования можно задать любые операции над процессом и спектром и реализовать практически любые сервисные функции.

Двухмашинный комплекс можно рассматривать как основу для построения сосредоточенных систем с общей памятью, в которых совокупность *двухмагистральных микроЭВМ* «Электроника С5-21М» образует однородную исполнительную часть. При этом резкое снижение интенсивности обращений модулей исполнительной части к общим ресурсам системы за счет *естественного отделения потоков локальных команд и данных от данных межпроцессорного обмена* позволяет определять производительность системы суммой производительностей отдельных микроЭВМ. Кроме того, использование нескольких однотипных микроЭВМ с центральным диспетчером (микроЭВМ «Электроника 60») обеспечивает повышение надежности (живучести) системы.

Статья поступила 27 марта 1984 г.

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Б. М. Малашевич, В. А. Шахнов, Э. И. Коночкин

Микропроцессорные средства и системы. Микропроцессорные интегральные микросхемы

Прошло более 10 лет со времени появления первых отечественных микропроцессоров и микроЭВМ, а единой общепринятой терминологии до настоящего времени, к сожалению, нет. Существующие нормативные документы имеют в основном отраслевое значение. Терминология в зарубежных источниках, тем более ее переводы на русский язык, также далеки от совершенства. Специализированного периодического издания, которое способствовало бы выработке единой терминологии, до выхода настоящего журнала не было.

В связи с этим в настоящей статье сделана попытка сформулировать предложения по единой терминологии в области микропроцессорных средств и систем. Авторы предполагают продолжить публикацию терминов и определений в последующих номерах журнала. Апробированная таким образом терминология будет использована в перерабатываемом отраслевом и планируемых к разработке государственном стандарте и стандарте СЭВ.

Ниже приводятся определения основных терминов и пояснения к ним, построенные на основе действующих нормативных документов*, общепринятых понятий и предложений авторов.

К ним относятся термины, составляющие вершину многоуровневого и широко разветвленного терминологического дерева (рис. 1, 2), построенного в соответствии с классификационной структурой микропроцессорных

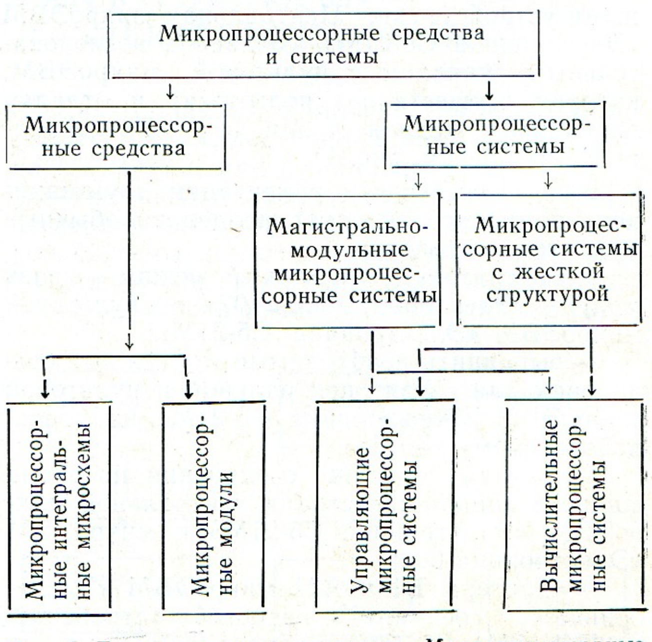


Рис. 1. Терминологическое дерево: «Микропроцессорные средства и системы»

* ГОСТ 1721-75 «Микросхемы интегральные. Термины и определения».

ОСТ 11 305.902-80 «Микропроцессорные средства вычислительной техники. Термины и определения».

средств и систем. За основу классификации взяты функциональные и конструктивные признаки. При определении и пояснении терминов часто используются другие термины. Те из них, которые последовательно определяются в данной статье, выделены курсивом.

Микропроцессорные средства и системы (МСС) — совокупность изделий вычислительной и управляющей техники и их функционально и конструктивно законченных составных частей, построенных на основе *микропроцессорных интегральных микросхем.*

Синонимами являются термины «микропроцессорная техника» и «микропроцессорные средства вычислительной техники» (МСВТ). Все эти термины не распространяются на машины, приборы и оборудование, в которых МСС используются, например, на измерительный прибор, на ЭВМ класса ЕС ЭВМ и т. п.

Микропроцессорные средства — конструктивно и функционально законченные изделия вычислительной и управляющей техники, построенные в виде или на основе микропроцессорных интегральных микросхем, которые, с точки зрения требований к испытаниям, приемке и поставке, рассматриваются как единое целое и применяются при построении более сложных микропроцессорных средств или *микропроцессорных систем.*

Конструктивно микропроцессорные средства выполняются в виде микросхемы, одноплатного изделия, моноблока или типового комплекса, причем изделия младшего уровня конструктивной иерархии могут использоваться в изделиях старшего.

Микропроцессорные системы — вычислительные или управляющие системы, построенные на основе микропроцессорных средств, применяются автономно или встраиваются в *управляемый объект.*

Конструктивно микропроцессорные системы выполняются в виде микросхемы, одноплатного изделия, моноблока, комплекса или нескольких изделий указанных типов, встроенных в аппаратуру управляемого объекта или выполненных автономно.

Управляемый объект — машина, прибор, аппарат, оборудование и т. п., управление работой которого осуществляется встроенной микропроцессорной системой.

Магистрально-модульная микропроцессорная система ($М^3С$) — микропроцессорная система, составные части которой объединяются на основе единого системного магистрального интерфейса и единой унифицированной конструкции.

Конструктивно могут выполняться на одной печатной плате (на основе однокристальных модулей или унифицированных фрагментов), в комплектном корпусе (объединение одноплатных модулей), в виде комплекса и т. п. Легко модернизируются, наращиваются и реконфигурируются.

Магистрально-модульная мультимикропроцессорная система — ($М^4С$) — магистрально-модульная микропроцессорная система, включающая несколько одинаковых или различных *микропроцессоров* или *микроЭВМ,* подключенных к одной системной магистрали и работающих под управлением одной операционной системы.

Микропроцессорная система с жесткой структурой — микропроцессорная система, составные части которой соединены неунифицированным образом.

Обладает меньшей избыточностью, но и меньшей гибкостью. Ее очень трудно или невозможно модернизировать, наращивать и реконфигурировать.

Управляющая микропроцессорная система (УМС) — микропроцессорная система, содержащая микроЭВМ, устройства связи с объектом (с датчиками и исполнительными органами управляемого объекта) и, при необходимости, периферийные устройства.

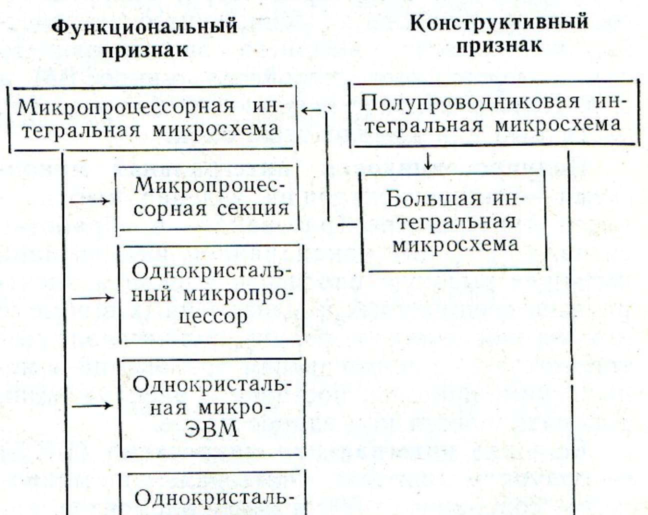


Рис. 2. Ветвь терминологического дерева: «Микропроцессорные средства и системы»

Может не иметь собственных источников электропитания и органов управления. Конструктивно, как правило, встраивается в аппаратуру управляемого объекта, работает в реальном масштабе времени.

Вычислительная микропроцессорная система (ВМС) — микропроцессорная система, содержащая микроЭВМ, внешние запоминающие устройства и периферийные устройства (в первую очередь дисплеи и печатающие устройства) и предназначенная для автоматизации рутинной части интеллектуального труда человека.

Выполняется в виде настольного комплекса или моноблока. Работает в режиме диалога с человеком и может выполнять задачи вычислений (инженерные, экономические и другие работы), подготовки документов (информационно-поисковые задачи), обработки результатов эксперимента и т. п.

Микропроцессорный модуль — конструктивно и функционально законченное изделие, имеющее унифицированные присоединительные характеристики (интерфейс, конструкция, программное обеспечение и т. п.).

Микропроцессорный модуль может выполнять функции: микропроцессора, оперативного или постоянного запоминающего устройства, контроллера, внешнего запоминающего или периферийного устройства, микроЭВМ и т. п. Конструктивное исполнение: микросхема, частичный или комплектный блок.

Полупроводниковая интегральная микросхема — микроэлектронное изделие, выполняющее функции преобразования и обработки сигнала и (или) накапливания информации, имеющее высокую плотность упаковки электрически соединенных и выполненных в объеме или на поверхности полупроводника элементов, которое, с точки зрения требований к испытаниям, приемке, поставке и эксплуатации, рассматривается как единое целое.

Большая интегральная микросхема (БИС) — полупроводниковая интегральная микросхема, содержащая 500 и более элементов, изготовленных по биполярной технологии, 1000 и более элементов, изготовленных по МДП технологии.

Микропроцессорная интегральная микросхема — интегральная микросхема, выполняющая функцию микропроцессора (микроконтроллера) или его части.

Микропроцессорная секция — микропроцессорная интегральная микросхема, реализующая часть микропроцессора (микроконтроллера) и обладающая средствами простого функционального объединения с однотипными или другими микропроцессорными секциями для построения законченных микропроцессоров, микроконтроллеров или микроЭВМ.

Однокристальный микропроцессор (ОМП) — микропроцессор, выполненный в виде одной большой интегральной микросхемы.

Однокристальная микроЭВМ (ОЭВМ) — микроЭВМ, выполненная в виде одной большой интегральной микросхемы.

Однокристальная микросистема — управляющая микропроцессорная система, выполненная в виде одной большой интегральной микросхемы.

Серия интегральных микросхем (серия) — совокупность типов интегральных микросхем, которые могут выполнять различные функции, имеют единое конструктивно-технологическое исполнение и предназначены для совместного применения.

Микропроцессорный комплекс БИС (МПК) — совокупность микропроцессорных и других интегральных микросхем, совместимых по архитектуре, конструктивному исполнению и электрическим параметрам и обеспечивающих возможность совместного применения.

Микропроцессорный набор — совокупность микропроцессорных и других интегральных микросхем микропроцессорного комплекта БИС, номенклатура и количество которых необходимы и достаточны для построения конкретного изделия вычислительной или управляющей техники.

Таким образом, понятия «серия», «комплект» и «набор» не тождественны. ***Серия*** — это производственное понятие, определяемое требованиями разработки, изготовления, испытания и поставки микросхем. В составе одной серии могут быть микросхемы, принадлежащие к различным комплектам. ***Комплект*** — номенклатурное понятие, объединяющее все микропроцессорные и сопутствующие им микросхемы с позиции возможности их совместного применения. При этом часто МПК заимствуют микросхемы из других комплектов или из серий стандартных ИС. ***Набор*** — прикладное, количественное понятие, объединяющее совокупность всех микросхем, применяемых для построения изделия.

Секционированный микропроцессорный комплект БИС — микропроцессорный комплект, состоящий из микропроцессорных секций.

Комплект однокристального микропроцессора — микропроцессорный комплект БИС, в котором каждое из основных функциональных устройств, включая микропроцессор, выполнено в виде одной большой интегральной микросхемы с встроенным системным интерфейсом.

РЕФЕРАТЫ СТАТЕЙ

УДК 681.3.06

Лавров С. С. **Использование вычислительной техники, программирование и искусственный интеллект (перспективы развития).** — Микропроцессорные средства и системы, 1984, № 1, с. 3.

Рассматриваются место и роль вычислительной техники в современном обществе и пути наиболее эффективного использования этой техники в интересах общества.

UDC 681.3.06

Lavrov S. S. Computer Application, Programming and Artificial Intelligence. — Microprocessor Devices and Systems, 1984, 3, p. 3.

The author considers the role of computers in the modern society and the ways of their most efficient application.

УДК 681.32:577.1

Дунин-Барковский В. Л. **Оценки возможностей биомолекулярных вычислений.** — Микропроцессорные средства и системы, 1984, № 3, с. 10.

Приводятся оценки эффективности работы клеточных систем в терминах вычислительной техники. Обсуждаются потенциальные возможности технического использования молекулярных биокомпонент.

UDC 681.32:577.1

Dunin-Barkovsky V. L. On the possibility of Biomolecular Computation. — Microprocessor Devices and Systems, 1984, N 3, p. 10.

The efficiency of cellular automata computer system is being analyzed. The potential technical use of biomolecular components is discussed.

УДК 681.32:577.155.3

Всеволодов Н. Н. **Применение «биохрома» для создания биокомпьютера.** — Микропроцессорные средства и системы, 1984, № 3, с. 13.

Показаны перспективы применения биологических микроустройств и фотобиоматериалов («биохромных» пленок) для создания биологической вычислительной машины — биокомпьютера.

UDC 681.32:577.155.3

Vsevolodov N. N. Using «Biochrome» for Biocomputers. — Microprocessor Devices and Systems, 1984, N 3, p. 13.

The article discusses the possibilities of using biological microdevices and photo-biomaterials («biohromatic films») for builging Biocomputer.

УДК 681.3.01:51

Хвощ С. Т., Смолов В. Б., Сухопаров А. И., Горовой В. В., Черняковский Д. Н., Чернуха Б. Н., Васильев А. Н. **Комплект БИС для организации мультиплексных каналов межмодульного обмена информацией.** — Микропроцессорные средства и системы, 1984, № 3, с. 18.

Рассматривается состав и архитектура комплекта БИС для организации протяженных высокопроизводительных мультиплексных каналов, предназначенных для комплексных систем сбора и обработки информации. Комплект БИС ориентирован на сопряжение мультиплексных каналов, работающих в коде «Манчестер-II», и общей шины ЭВМ «Электроника 60М». Приводятся примеры, иллюстрирующие специфику применения БИС комплекта.

UDC 681.3.01:51

Khvosch S. T., Smolov V. B., Sukhoparov A. I., Gorovoy V. V., Tchernyakovsky D. N., Tchernukha B. N., Vasiliev A. N. IC set for Multiplexed Channels of Inter module Information Exchange. — Microprocessor Devices and Systems, 1984, N 3 p. 18.

The architecture of Ic set Is set considered which is intended for building distant high-efficient multiplexed channels. Such channels are used in complex data acquisition data processing systems. The set provides interfacing of multiplexed channels using «Manchester-II» codes with «Electronica-60M» microcomputer bus sustem.

УДК 621.3.049.77:681.325.5

Бобков В. А., Бобровницкий М. М., Горовой В. В., Евдокимов В. А. **Микропроцессорный комплект БИС КА1808 и контроллер зеркальных фотокамер на его основе.** — Микропроцессорные средства и системы, 1984, № 3, с. 23.

Рассмотрены архитектура, основные характеристики всех БИС комплекта КА1808 и контроллера на его базе, а также особенности применения БИС.

UDC 621.3049.77:681.325.5

Bobkov V. A., Bobrovnitcky M. M., Gorovoi V. V., Yevdokimov V. A. KA1808 Microprocessor Set and Controller for Reflection-type Photocameras. Microprocessor Devices and Systems, 1984, N 3 p. 23.

The architecture and specification of KA1808 IC-s set are considered. Their application for building reflection-type photocameras are described.

УДК 681.3.002

Козак А. А., Сорока С. И. **Устройства отображения информации микропроцессорных средств вычислительной техники.** — Микропроцессорные средства и системы, 1984, № 3, с. 27.

Уточняются терминология и основные понятия в области устройств отображения информации микропроцессорных средств вычислительной техники (УОИ МСВТ). Рассмотрено современное состояние, тенденции и перспективы развития дисплеев МСВТ. Приведена классификация УОИ МСВТ, описаны некоторые модели, указаны их основные технические характеристики.

УДК 681.324

Мячев А. А. **Принципы согласования системных интерфейсов мини- и микроЭВМ.** — Микропроцессорные средства и системы, 1984, № 3, с. 32.

На основе анализа тенденций развития систем, использующих технические средства различных ЭВМ, сформулирована концепция нового класса подсистем ввода-вывода — устройств схемного согласования системных интерфейсов (УСИ) микро- и мини-ЭВМ и микропроцессорных систем. Рассмотрен широкий круг вопросов, охватывающих наиболее важные аспекты структурных, технических и методологических проблем организации УСИ и направления их развития.

УДК 681.3.06:681.322.1

Брябрин В. М. **Интеллектуальный интерфейс на основе персональной ЭВМ.** — Микропроцессорные средства и системы, 1984, № 3, с. 38.

Рассматриваются методы организации человеко-машинного взаимодействия, основанного на использовании персональных ЭВМ. Излагается новый подход, в соответствии с которым интеллектуальный интерфейс пользователя с прикладной системой обеспечивает многообразие форм человеко-машинного взаимодействия, единство и естественность стиля работы пользователей, интеграцию представления данных и унификацию связей с пакетами прикладных программ. Основные программные компоненты системы — объектная база, управление дисплейными окнами, графический интерфейс, текстовый процессор и поддержка адаптивной диалоговой среды. В статье излагается структура и функционирование отдельных компонентов интеллектуального интерфейса. Кратко рассматриваются прикладные системы, построенные на основе данного подхода.

УДК 681.3.06

Кыпп М., Тыугу Э. **Объектно-ориентированная система программирования для малых ЭВМ.** — Микропроцессорные средства и системы, 1984, № 3, с. 44.

Описана система программирования Микроприд, написанная на языке Паскаль для микроЭВМ Apple II. Упрощенный входной язык и экономный внутренний язык делают эту систему доступной для непрофессиональных программистов.

УДК 61:658

Цейтин Г. С. **Психологические задачи в разработке программного обеспечения.** — Микропроцессорные средства и системы, 1984, № 3, с. 48.

Рассмотрены психологические факторы, существенно влияющие на эффективность работ по созданию программного обеспечения. Составлена классификация психологических задача, возникающих в ходе разработки программ. Выделено шесть классов, объединяющих 37 задач со множеством подзадач.

UDC 681.3.022

Kozak A. A., Soroka S. I. Display devices for microcomputers. — Microprocessor Devices and Systems, 1984, N 3, p. 27.

The terminology and basic concepts are formulated; the state-of-the-art, trends and perspectives of the display devices development are discussed. A classification is given, some models are described.

UDC 681.324

Miachev A. A. The principles of mini-micro computer systems agreement. — Microprocessor Devices and Systems, 1984, N 3, p. 32.

The evolution of systems involving devices of different computers is analyzed. The devices for system interface agreement are considered as a new class of input/output devices. Structural, technical and methodological aspects of their organization are discussed.

UDC 681.3.06:681.322.1

Briabrin V. M. Intelligent Interface for Personal Computer. — Microprocessor Devices and Systems, 1984, N 3, p. 38.

The paper consideres the methods of man-computer interaction with the personal computers. The new approach for building intelligent interface erovides the diverse forms of man-computer interaction, the naturalness of user's style, integrity of data and standardization of application access. The system consists of the following basic components: object base, window manager, graphical interface, text processor and adaptive dialogue support. The structure and functional features of these components are discussed and the brief description of applications is presented.

UDC 681.3.06

Kypp M., Tyugu E. Object-Oriented Programming System for Small Computers. — Microprocessor Devices and Systems, 1984, N 3, p. 44.

Micro Priz is an object-oriented programming system written in Apple Dos Pascal. The simplified input language and optimized internal structures make the system easy to use by non-professional programmers.

UDC 61:658

Tceitin G. S. Human Factors in Software Development. — Microprocessor Devices and Systems, 1984, N 3, p. 48.

The human factors of software development are considered. The classification of psychological problems is proposed which qrise in the software development process. 37 problems are grouped into 6 classes, each problem consisting of several subproblems.

УДК 681.3

Иванов Е. А., Муренко Л. Л., Широков Ю. Ф. **Универсальная отладочная система автоматизации проектирования микропроцессорных устройств «Электроника ТЗ».** — Микропроцессорные средства и системы, 1984, № 3, с. 53.

Рассматриваются принципы построения универсальной отладочной системы для разработки микропроцессорных устройств «Электроника ТЗ», требующей минимальных ресурсов для своего производства и обеспечивающей возможность построения отладочных средств для различных типов микропроцессоров и однокристальных микроЭВМ. Приводится классификация отладочных средств, описывается структура и характеристики отладочного комплекса, построенного на основе ДВК «Электроника НЦ-80—20/2», аппаратных и программных модулей (схемные эмуляторы, программаторы БИС ППЗУ и ПЛМ, пакет отладочных программ).

UDC 681.3

Ivanov E. A., Murenko L. L., Shirokov Yu. F. Universal Debugging System for Automated Design of «Electronica-T3» Microprocessor Devices. — Microprocessor Devices and Systems, 1984, N 3, p. 53.

The principles of universal debugging system are considered with application to automated design of «Electronica-T3» series of microprocessor devices. The system requires limited resources and provides the development of debugging systems for different types of microprocessors and controllers. The classification of debugging systems is proposed. The structure and specification of «Electronica NC-80-20/2» — system is described which includes logical emulators, PROM and PAL programmers and appropriate software.

УДК 681.325.5—181.48.004:621.39

Кудрявцев Г. Г., Мамзелев И. А. **Микропроцессорные средства в технике связи.** — Микропроцессорные средства и системы, 1984, № 3, с. 60.

Обсуждаются вопросы использования микропроцессорных средств вычислительной техники в аппаратуре обмена и распределения информации — от сложных систем коммутации до телефонного аппарата. Приводятся примеры действующих и разрабатываемых систем. Дается обзор применений микропроцессорной техники в почтовой связи и автоматизированных системах технической эксплуатации. Описывается система связи ближайшего будущего.

UDC 681.325.5—181.48.004:621.39

Kudriavtsev G. G., Mamzelev I. A. Microprocessor devices in communication. — Microprocessor Devices and Systems, 1984, N 3, p. 60.

The application of microprocessor in different information exchange/distribution devices, from a telephone to complex commutation systems. Some examples are given. The microprocessor applications in post communication and automated systems are reviewed. A communication system of near future is described.

УДК 007.52

Чернухин Ю. В., Носков В. П., Каляев Н. А., Усачев Л. Ж., Мишкинюк В. К., Сологуб П. С. **Микропроцессорная система управления транспортного робота.** — Микропроцессорные средства и системы, 1984, № 3, с. 70.

Рассмотрены возможности реализации системы управления транспортного робота с применением микропроцессорной техники. Блоки формирования модели внешней среды, задания цели и навигации могут быть реализованы на основе микроЭВМ, процессор — в виде однородной структуры, состоящей из 448 ячеек (микросхем серии К164).

UDC 007.52

Tchernukhin Yu. V., Noskov V. P., Kalyaev N. A., Usatchgov L. J., Mishkinyuk V. K., Sologub P. C. Microprocessor Control System for Transportation Robot. — Microprocessor Devices and Systems, 1984, N 3, p. 70.

The possibilities of building microprocessor-based control systems for transportation robot are considered. Microcomputer provides manipulation of the world model and navigation goals. The processor consists of 448 cells (K144 IC set) arranged into uniform computational structure.

УДК 621.3.049.77

Давыдов Н. П., Давыдова Ж. К., Мишин Г. Т., Савченко А. П., Чернушенко А. М., Язовцев В. И. **Система управления технологическими процессами на основе микропроцессорного комплекта БИС серии КР580.** — Микропроцессорные средства и системы, 1984, № 3, с. 72.

Анализируются аппаратное и программное обеспечение системы автоматического управления технологическим процессом изготовления прецизионных тонкопленочных микрорезисторов типа АУТИМ, построенной на базе микропроцессорного комплекта БИС КР580.

UDC 621.3.049.77

Davydov N. P., Davydova J. K., Mishin G. T., Savtchenko A. P., Tchernushenko A. M., Yazovtcev V. I. Computer Aided Manufacturing System built on KP580 IC Set. — Microprocessor Devices and Systems, 1984, N 3, p. 72.

The CAM system is described which is intended for production of precision thin-film microresistors. The system is built around the KP580 IC set.

Пятиугольник «МП»

В редакцию журнала приходят письма, авторы которых, в основном преподаватели и сотрудники технических вузов, предлагают для публикации статьи, содержащие обильно насыщенные греческими и латинскими символами формализованные описания, по их мнению, крайне полезные для широкого круга специалистов в области разработки и внедрения микропроцессорных систем. В процессе возникающей иногда вслед за отклонением таких статей бурной переписки, авторы высказывают свое недоумение, в частности, и тем обстоятельством, что некоторые члены редколлегии и редсовета «МП», известные ученые, статьи которых (нередко написанные на недосягаемо формальном уровне) печатаются в ведущих советских и зарубежных научных изданиях, тем не менее в свой журнал «МП» предпочитают писать ярким образным языком, четко излагая наиболее сложные концепции развития микропроцессорной техники.

Учитывая эти письма и замечания, редакция считает целесообразным в дальнейшем не реже, чем один раз в два-три года публиковать статьи, выдержанные в традиционном для некоторой части межвузовских политехнических сборников «греко-латинском» стиле.

Предлагаем вниманию читателей первую из статей этого цикла.

Из письма в редакцию: «...Тезисы статьи были опубликованы в сборнике трудов XXIV межвузовской конференции «АСУ—ФАП—ГАП». В ее основе лежит отчет, выполненный в рамках хоздоговорной тематики по микропроцессорным основам символьного исчисления для зубофрезерного АСУТП в условиях массового домостроения и мелкосерийного производства».

Д. О. Цент

ЭЛЕКТРОННЫЙ КЛЮЧ В ФУНКЦИОНАЛЬНО ПОЛНОЙ СИСТЕМЕ КЛЮЧЕЙ: ВОПРОСЫ МАТЕМАТИЧЕСКОГО ОБЕСПЕЧЕНИЯ

В настоящем сообщении мы излагаем упрощенную теоретическую схему, на основе которой в будущем предполагается создавать развитые системы математического обеспечения для механических агрегатов современных жилых зданий со встроенными микропроцессорами и микроЭВМ.

Начнем с определений. Управляемый командами встроенной микроЭВМ ключ состоит из стержня, на котором укреплены штифты. Замок состоит из щели с отверстиями, расположенными соответственно позициям штифтов на стержне ключа. Кроме того, в замке имеется система рычажков, находящихся позади отверстий.

Введем следующие три аксиомы:
— штифты поворачивают рычажки. Для того, чтобы замок открылся, все рычажки в замке должны быть повернуты;
— если в данной позиции нет штифта, отверстия или рычажка, мы будем говорить в дальнейшем о наличии в данной позиции антиштифта, антиотверстия или антирычажка соответственно;
— ни в одном замке нет рычажков за антиотверстиями, ибо такой замок в рамках известных ограничений нерелятивистской механики нельзя было бы открыть.

Пусть штифты, отверстия и рычажки описываются значением 1 переменных a_i, b_i, c_i соответственно (индекс i — номер позиции). Антиштифты, антиотверстия и антирычажки соответствуют значению 0 тех же переменных. Определим теперь матричное умножение следующим способом:

$$(a_1\, a_2 \dots a_k)\begin{pmatrix} b_1 & b_2 & \dots & b_k \\ c_1 & c_2 & \dots & c_k \end{pmatrix} = (a_1\, b_1\, c_1,\ a_2\, b_2\, c_2,\ \dots,\ a_k\, b_k\, c_k), \qquad (1)$$

где символическое произведение $abc = a$, если одновременно $a \leqslant b$ и $a \geqslant b$, в противном случае $abc = 1 - a$. Отсюда следует, что если $(a_1 a_2, \dots, a_k)$ есть собственный вектор оператора

$$\begin{pmatrix} b_1 & b_2 & \dots & b_k \\ c_1 & c_2 & \dots & c_k \end{pmatrix},$$

то ключ может отпереть замок. (Весом корпуса и диаметром дужки навесного замка можно пренебречь).

Используя этот формализм, легко найти полное число электронных ключей, которые открывают данный навесной механический замок $\begin{pmatrix} b \\ c \end{pmatrix}$. Оно равно

$$N_k = (2^{\Sigma(b_i - c_i)}) \cdot k \qquad (2)$$

а число замков, которые могут быть открыты данным ключом (a), равно

$$N_L = (2^{\Sigma a_i} - 1) \cdot k. \qquad (3)$$

При получении этих выражений учитывался тот факт, что замок $\begin{pmatrix} o \\ o \end{pmatrix}$ есть тривиальный антизамок. В уравнениях (2) и (3) k есть сумма коэффициентов Клебша — Гордана, равная единице.

Развитый выше формализм позволил в частности решить следующую практическую задачу. Пусть некто хочет пройти (например, используя микроЭВМ с голосовым вводом пароля), из некоторой комнаты A через несколько дверей в произвольную комнату B. Число электронных ключей, необходимое для этого, максимизировалось при произвольном выборе комнат A и B. (Проблема минимизации не решалась, поскольку ее решение тривиально — одинаковые замки). Затем сотрудники института были разбиты на ряд подгрупп, и кодовая система электронных ключей строилась таким образом, чтобы одновременно выполнялись два условия:

— ни одна подгруппа не в состоянии открыть все те замки, которые могут быть открыты любой другой подгруппой;

— трансформационные свойства групп соответствуют возможности одалживания механических ключей, а также вероятности разгадывания (похищения) пароля для микропроцессорных реализаций.

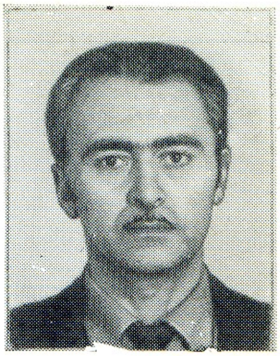
Создатели математической системы электронных ключей с профилирующей кафедры сначала надеялись, что она является единственно возможной и полной, и до известной степени это справедливо. Однако при экспериментальных исследованиях оказалось, что ключи, которые не должны были бы открывать некоторые двери, открывают их, если их вставлять в замок не до конца. Например, ключ (11111) может открыть замок $\begin{pmatrix} 10000 \\ 11111 \end{pmatrix}$ в $n = 5$ различных положениях. Число n было названо «странностью системы ключ — замок». Экспериментальными исследованиями было установлено, что наша система ключей является весьма странной. Однако этот недостаток можно исправить, если потребовать для последней позиции соблюдения равенств $a_k = b_k = c_k = 1$. Будем надеяться, что при ближайшем пересмотре Госстандартом системы ключей в нее будет внесено это предлагаемое нами исправление.

Границы применимости: на отмычки настоящее исследование не распространяется.

Автор выражает благодарность сотрудникам факультета, работающим на смежных кафедрах, за горячее обсуждение затронутых проблем и помощь в проведении экспериментальных исследований, а также сотрудникам ВЦ института за регулярно предоставляемый нам приоритет в пакетной обработке растущего потока диссертационных задач, вытекающих из предлагаемого выше формализма.

Из справки о внедрении: «...Общая сумма регионально-экономической эффективности данной хоздоговорной работы в пересчете на сезонно-отраслевой коэффициент оценивается согласно методике Х.Н.О. в 137 млн. руб. С учетом разницы поясного времени и почасовой тарифной сетки это составляет в среднем по факультету — 486,7 млн. руб. на одну справку о внедрении».

КНИГИ ПО МИКРОПРОЦЕССОРНОЙ ТЕХНИКЕ ИЗДАТЕЛЬСТВА «РАДИО И СВЯЗЬ»



Калугин Игорь Константинович
Главный редактор издательства «Радио и связь»

В тематических планах издательства «Радио и связь» значительное внимание уделяется книгам по микропроцессорной технике. Учитывая особую важность этого тематического направления, в 1984 г. разработан план выпуска книг по микропроцессорным средствам и системам.

В соответствии с этим планом в 1984 г. выпущена книга А. В. Каляева «Многопроцессорные системы с программируемой архитектурой». Автор поставил задачу — ознакомить широкий круг читателей с принципами построения и состоянием развития нового класса многопроцессорных вычислительных систем с программируемой архитектурой, универсальной коммутацией и распределенной памятью, основу которых составляют микропроцессоры (МП), работающие с машинными языками высокого уровня и реализующие достаточно сложные операции путем программирования структуры, а не процедуры, в отличие от традиционных МП.

В этом году издательством выпущена также книга А. Г. Алексенко, А. А. Галицына и А. Д. Иванникова «Проектирование радиоэлектронной аппаратуры на микропроцессорах». На основании отечественного и зарубежного опыта, в том числе работ, проводимых под руководством авторов, излагается методика проектирования цифровой аппаратуры на МП, причем главное внимание уделяется разработке технических средств и программного обеспечения для сопряжения микроЭВМ с управляемым или контролируемым объектом, ЗУ систем на МП, комплексной отладке МП систем. Подробно рассматривается процесс проектирования терминальных устройств, систем с числовым программным управлением, контрольно-измерительной аппаратуры и цифровых фильтров на МП, а также приводятся примеры проектирования.

В 1984 г. выйдет еще ряд книг по микропроцессорной тематике. Инженерам, занимающимся разработкой и применением микропроцессорных комплектов больших интегральных схем (МПК БИС), адресуется книга коллектива авторов под редакцией Э. П. Калошкина «Микропроцессорные комплекты БИС на основе интегральной инжекционной логики». В ней рассматриваются составы МПК БИС, анализируются логические структуры, описываются основные характеристики и система параметров, формулируются принципы построения вычислительных устройств и систем управления на базе МПК БИС серий К583 и К584, приводятся соответствующие примеры, отмечаются тенденции и перспективы дальнейшего развития рассматриваемых МПК БИС.

Можно надеяться, что инженеры-разработчики аппаратуры на основе МПК БИС с интересом встретят книгу С. С. Булгакова, В. М. Мещерякова, Л. А. Шумилова и др. «Проектирование цифровых систем на комплектах микропрограммируемых БИС». С единых методологических позиций рассматривается комплекс вопросов, связанных с проектированием и технической реализацией цифровых систем на современной элементной базе, предлагаются способы генерации и оценки проектных решений на структурном и алгоритмическом этапах проектирования систем. В книге будет подробно описан новый МПК БИС К1804.

Среди книг по МП технике, выпускаемых в этом году, есть и научно-популярное издание. Это книга Г. Я. Мирского «Микропроцессоры в измерительных приборах». В популярной форме даются общие сведения о МП, применении их в электронных осциллографах, цифровых вольтметрах, частотомерах. Автор кратко рассматривает приборные интерфейсы и общие принципы построения измерительно-вычислительных комплексов, раскрывает преимущества и новые возможности, получаемые от использования МП в измерительной аппаратуре.

Микропроцессорная тематика в той или иной мере затрагивается и в других книгах выпуска 1984 г. В качестве примеров здесь можно назвать такие издания, как «Аналоговые и цифровые интегральные микросхемы. Справочник» под ред. С. В. Якубовского, «Проектирование микросхем на элементах с инжекционной логикой» авторов И. И. Шагурина и К. О. Петросянца и др.

Из числа книг по данной тематике, планируемых к изданию в 1985 г., следует отметить совместную работу коллектива авторов из СССР и ГДР (Е. В. Бойченко, В. Кальфа, В. В. Овчинников) «Локальные вычислительные сети». В книге освещаются вопросы проектирования недорогих рассредоточенных вычислительных комплексов, объединяющих МП и микроЭВМ, встроенные в соответствующее управленческое и конторское оборудование.

Внимание специалистов должны привлечь также книги коллектива авторов под ред. Б. Н. Файзулаева и Б. В. Тарабрина «Применение интегральных микросхем в ЭВМ», В. Д. Байкова и В. Б. Смолова «Специализированные процессоры: Итерационные алгоритмы и структуры» и др.

Ряд книг по МП технике предполагается выпустить в 1986 и 1987 гг. Своеобразным дополнением к справочнику коллектива авторов под ред. А. А. Васенкова и В. А. Шахнова «Микропроцессорные комплекты интегральных схем» (издательство «Радио и связь», 1982 г.) явится работа В. Б. Абрайтиса, С. Ю. Седаускаса и А. В. Пятраускаса «Микропроцессорный комплект БИС высокого быстродействия К1800». В ней будут приведены структурные, функциональные схемы МП, описание принципов работы, указаны электрические и эксплуатационные параметры.

С интересом, по-видимому, будет встречена и книга Г. К. Йена, К. Г. Бичема и Г. П. Робинсона «Микропроцессорные системы и их применение в системах обработки сигналов» (перевод с английского). В ней описывается устройство МП, правила их применения, вопросы программирования, а также иллюстрируются возможности применения МП при обработке сигналов, вопросы же самой цифровой обработки сигналов излагаются в минимально необходимой степени. Существенным достоинством этой работы является то, что рассматриваемые в ней МП совместимы с освоенными у нас в стране, а это позволяет нашим специалистам использовать ряд рекомендаций авторов.

Инженерам, занимающимся разработкой как самих МП, так и различных систем на их основе, адресуются книги П. О. Видениекс и др. «Проблемно-ориентированные микропроцессорные системы в производстве РЭА»; В. М. Корячко «Конструирование микропроцессорных систем контроля РЭА» и др.

Планируется выпустить несколько книг для широкого круга читателей. Здесь прежде всего следует сказать о работе А. В. Шилейко и Т. И. Шилейко «Микропроцессоры». Интересен подход, используемый авторами при разработке данной темы — материал излагается от условий появления МП, от положений теории алгоритмов и техники интегральных схем с переходом к БИС с программно-перестраиваемой логикой, собственно МП и их применению в микроконтроллерах и микроЭВМ. Кроме того, сам МП преподносится читателю не как специальный элемент вычислительной техники, а как универсальный элемент автоматики с чрезвычайно обширной сферой применений. Думается, что эта книга вызовет интерес у читателей самых различных категорий.

Для массового читателя готовятся издания авторов И. В. Варламова, И. В. Касаткина «Микропроцессоры в бытовой технике»; П. В. Зеленко, В. В. Панова, С. Н. Попова «Домашний компьютер». Эти книги выйдут в серии «Массовая радиобиблиотека».

В выпускаемой издательством «Радио и связь» литературе по микропроцессорной технике нетрудно заметить некоторый уклон в сторону освещения принципов действия МП, их проектирования, технологии изготовления. Но ведь специалисты с нетерпением ждут работ, касающихся применений МП в конкретных областях. Видимо, назрела необходимость издания фундаментального справочника по МП, а также других книг, которые в основном имели бы прикладной характер.

1р. 10к.

Индекс 70588

